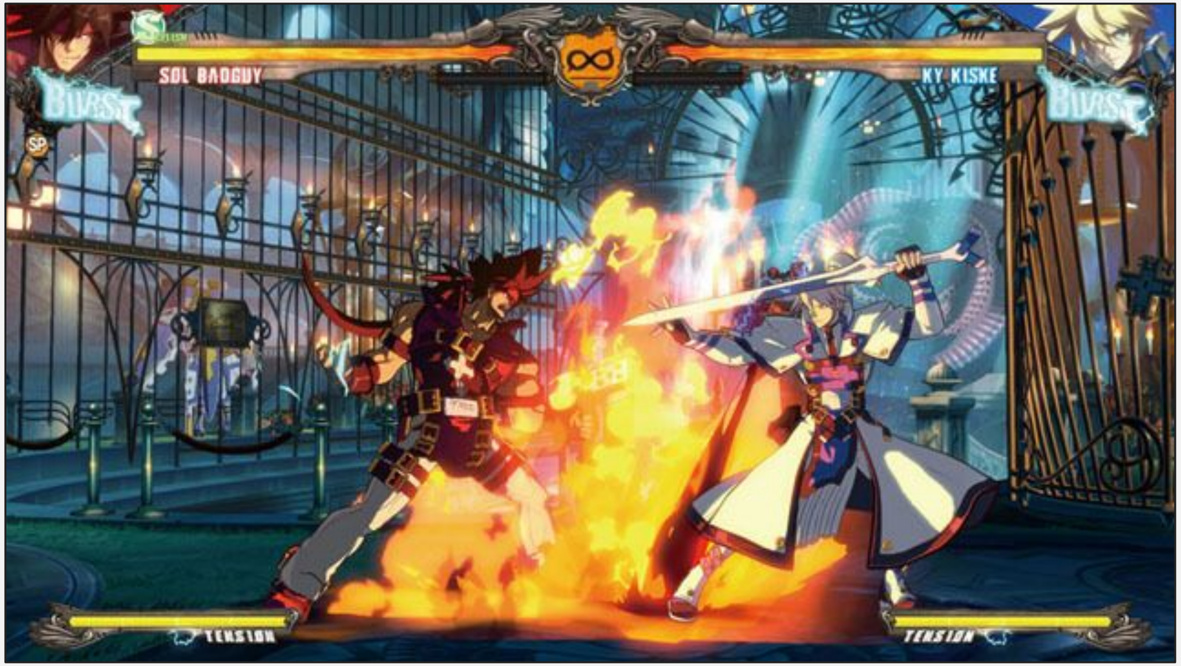
CGWORLD

Non-photorealistic **GUILTY GEAR Xrd -REVELATOR-** has evolved even further by adding features unique to 3D.



This series has been fascinating players with its 2D anime-like visuals and it's even more impressive with its latest instalment, GUILTY GEAR Xrd -REVELATOR-. This time, let's focus on elements other than the ones we've previously covered (CGWORLD vol.214)

ORIGINAL ARTICLE: TEXT_武田かおり/ KAORI TAKEDA, CGWORLD Vol.215
EDIT__藤井紀明 / Noriaki Fujii（CGWORLD）、山田桃子 / Momoko Yamada

Translation and slides by @LeoBGKK on Twitter

Improving the visuals without harming gameplay

GUILTY GEAR Xrd -REVELATOR- is the sequel to GUILTY GEAR Xrd -SIGN-, a fighting game that drew a lot of attention for its 3DCG recreation of 2D anime-like visuals.

Development began in December 2014, after the release of the previous title, and after the arcade version was released in August 2015, the console version is coming out next. “At Arc System Works we have a development plan that extends several years into the future, releasing brushed-up sequels as new titles," said Daisuke Ishiwatari, who has served as general director throughout the series.

From the left:

Kayumi Takuro(logicalbeat), lead programmer

Kazuya Igarashi, background artist

Daisuke Ishiwatari, director

Junya C. Motomura, technical artist

Eiji Kato, background artist

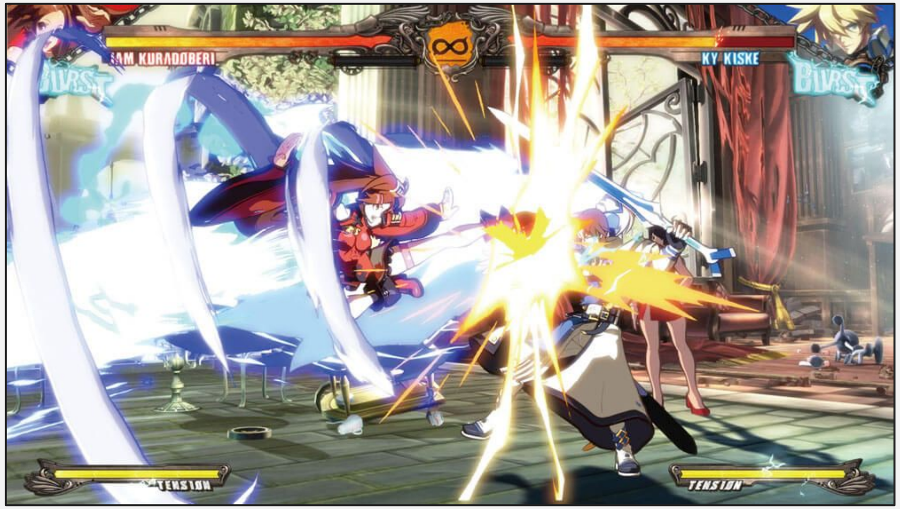
Hidehiko Sakamura, art director and lead animator

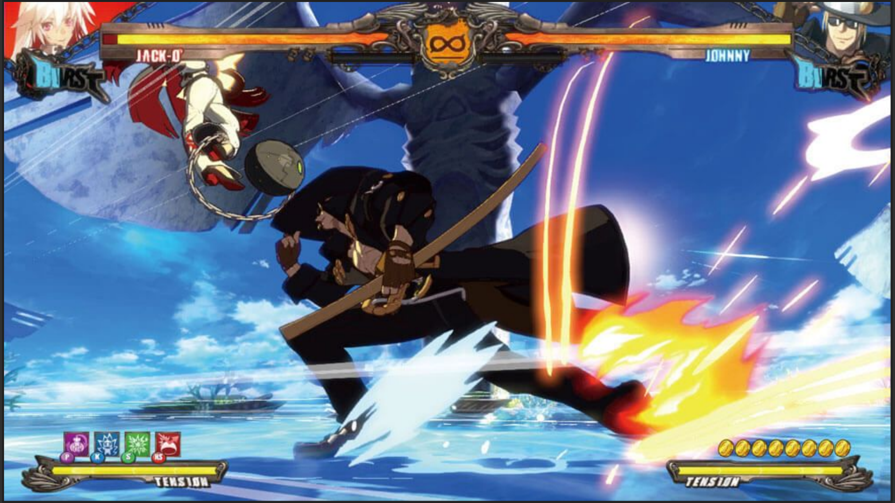
Genki Mamada(ArcSystemWorks), lead background artist

The visual concept of this series is to recreate 2D anime images, but to what extent will 3DCG be acceptable for the anime look?

According to Junya Motomura C., technical artist, that line between the two was not drawn well in the previous title, and there were parts that were deliberately suppressed. Specifically, in the previous game, the character shading was fixed for all stages, in line with traditional fighting games, but *"we were able to get feedback from the users on the acceptable range of 3D,"* said Hidehiko Sakamura, art director and chief animator.

For example, in this work, characters are affected by light from the environment to the extent that it does not affect their playability.





The core staff consists of 12 designers and 5 programmers. The workflow was already established in the previous game, but since the story stage backgrounds in this game are almost full 3DCG and there are many of them, they had to review them many times.

"For this game, we created a new extension tool for the story mode, which allows you to manipulate parameters in the Unreal Engine from an external editor," said Takuro Ieyumi, lead programmer. With this tool, we can control events in the background with scripts, which has broadened the scope of production and increased the production speed.

Now, let's take a look at the visual breakdown of this work, which has been further brushed up from the previous work.

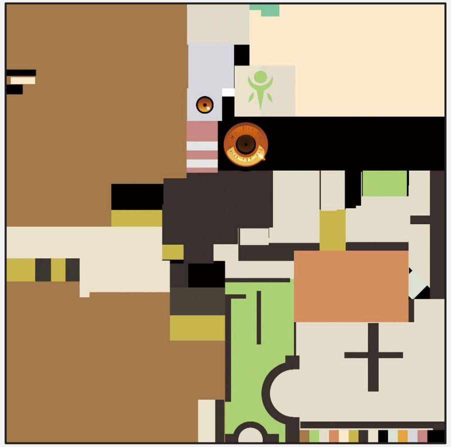
/01/
Character Rendering

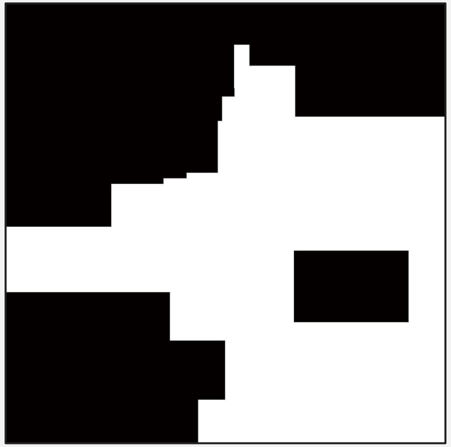
One of the major changes from the previous game is the reflection of rim light, point light, and environmental light of each stage on the characters.

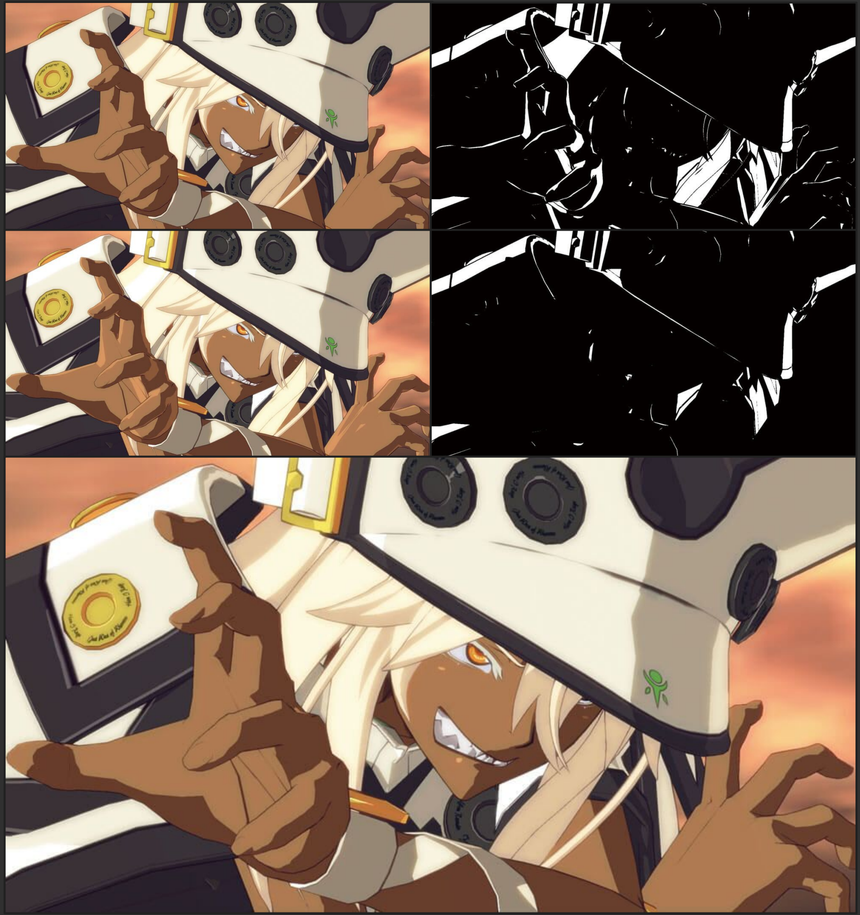
Also, battle damage has been added to the characters.

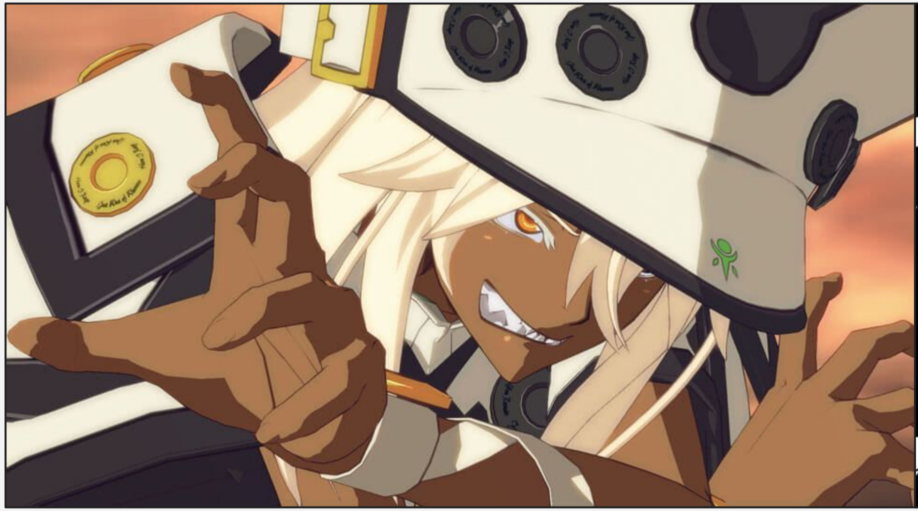
Improved visuals with rim lighting

In the previous game, the highlighting and shadowing of the characters were the same in both close-ups and far away, so there was a disparity in the amount of information. This time, rim lighting is added to control the amount of information for each image.

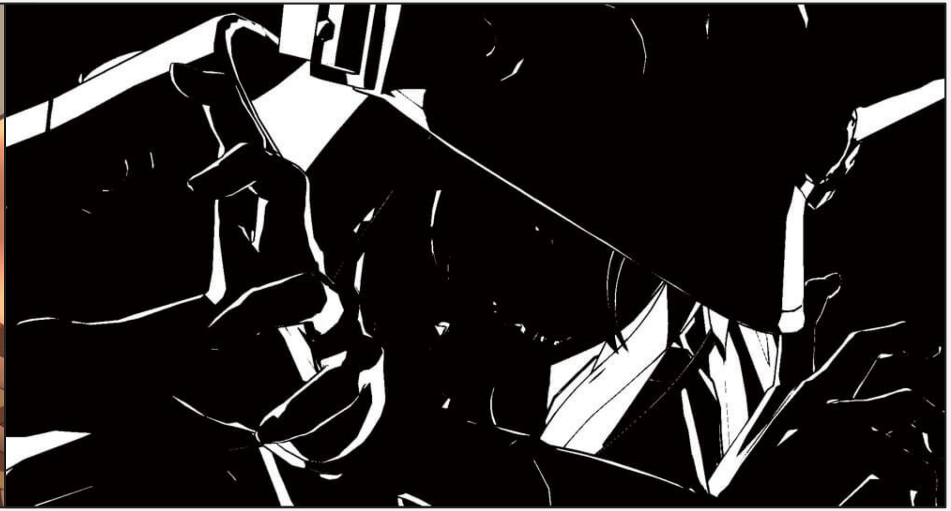

↑ Base Texture


↑ Mask for the rim light, stored in the alpha channel of the Base Texture.

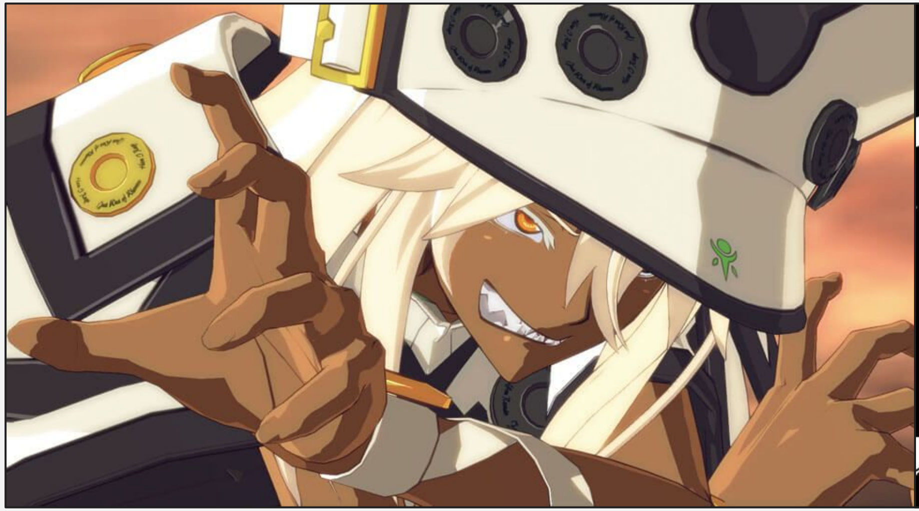


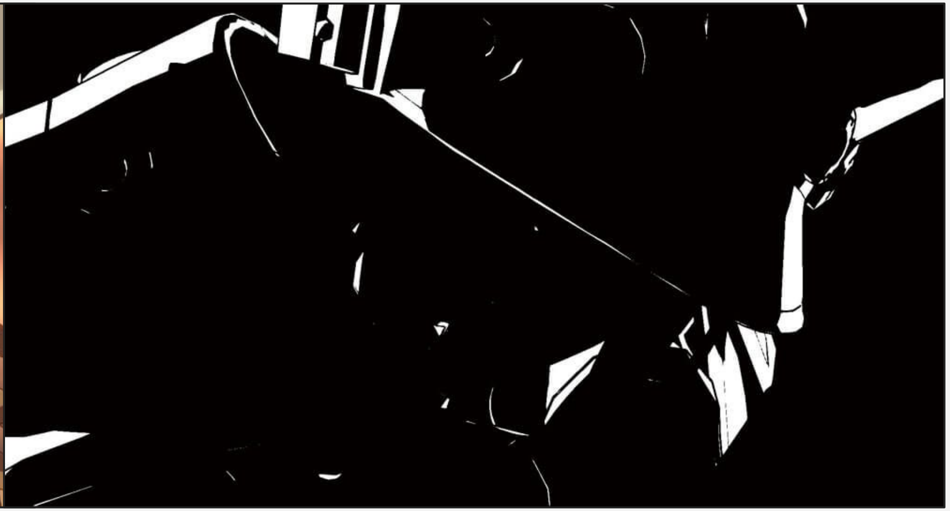


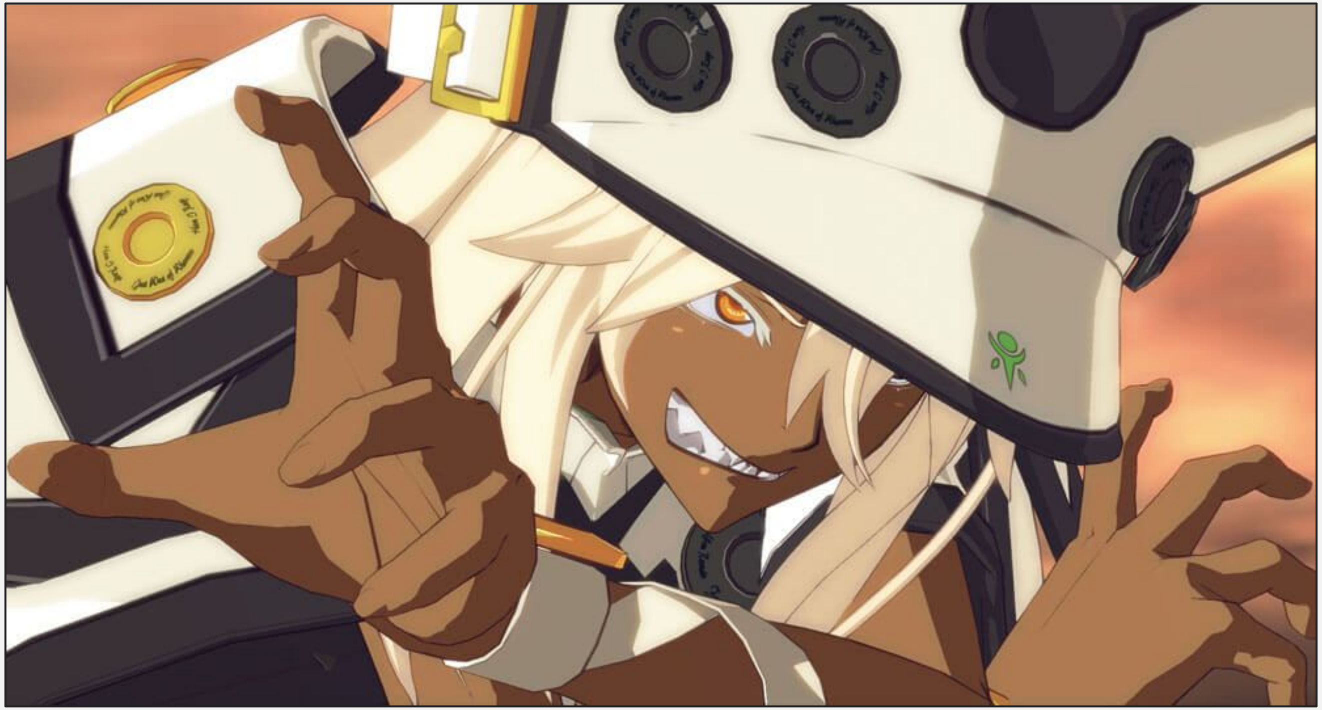
↑ Rim Light Off



↑ Rim Light Only


↑ Rim Light ON. The color is emphasized here for readability.


↑ Rim Light Only after masking via texture map. Removed areas where we don’t want to see a rim light, such as the hair.

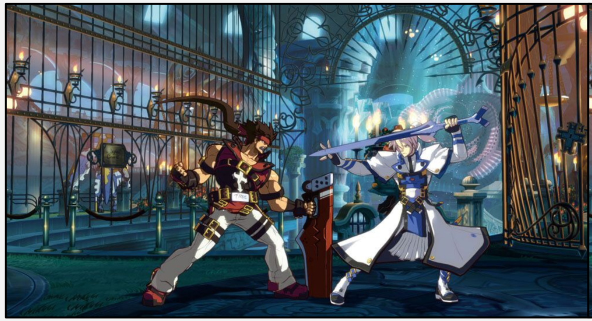


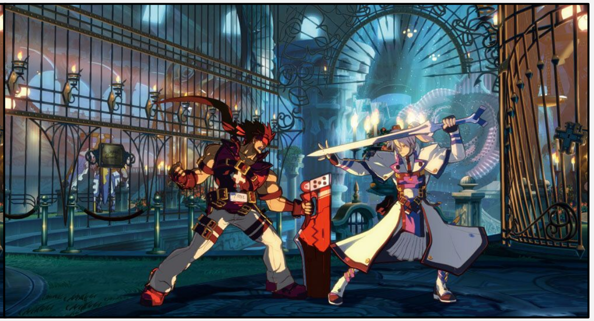
↑ Final image. Compared to the image without a rim light, the style of the character is much better.

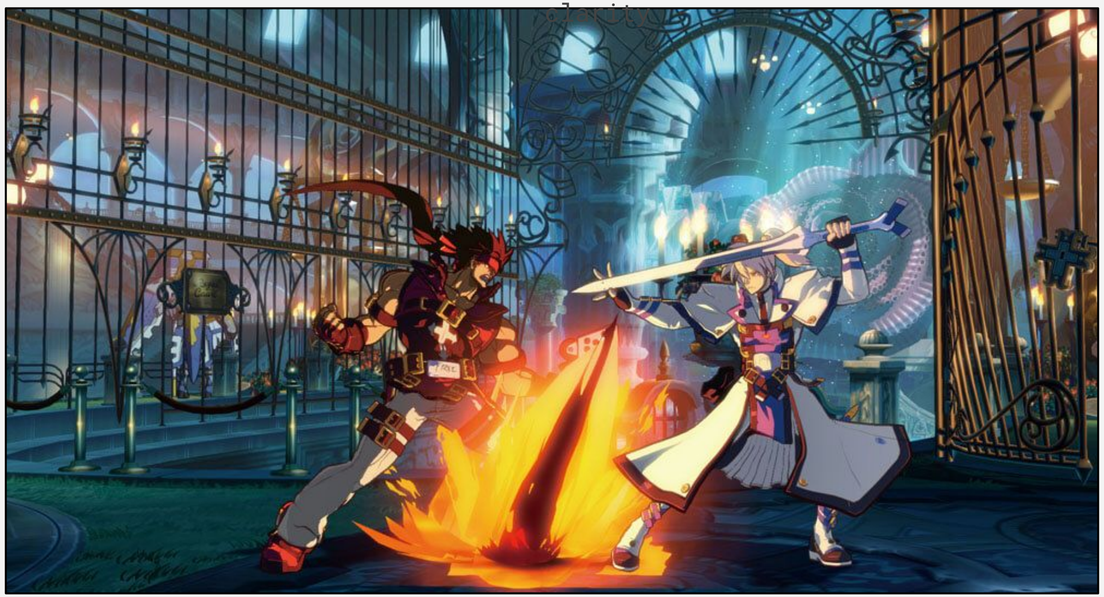
Reflecting other lights

Characters in the game are now affected by point lights from attacks and environments, and their colors change.

These point lights are not **added** to the environmental lights, but rather, are **replaced based on distance**. The closer the point light gets to the character, the more the ambient light is taken over by the point light and the more it affects the character. In normal circumstances, we would add point lights, but that would make them too bright and reduce visibility. *"It was always a challenge to find the right balance between appearance and gameplay."* (Mr. Motomura). Having a system in which the character's color changes under the influence of the light greatly increases the realism of the game.


↑ Point Light OFF

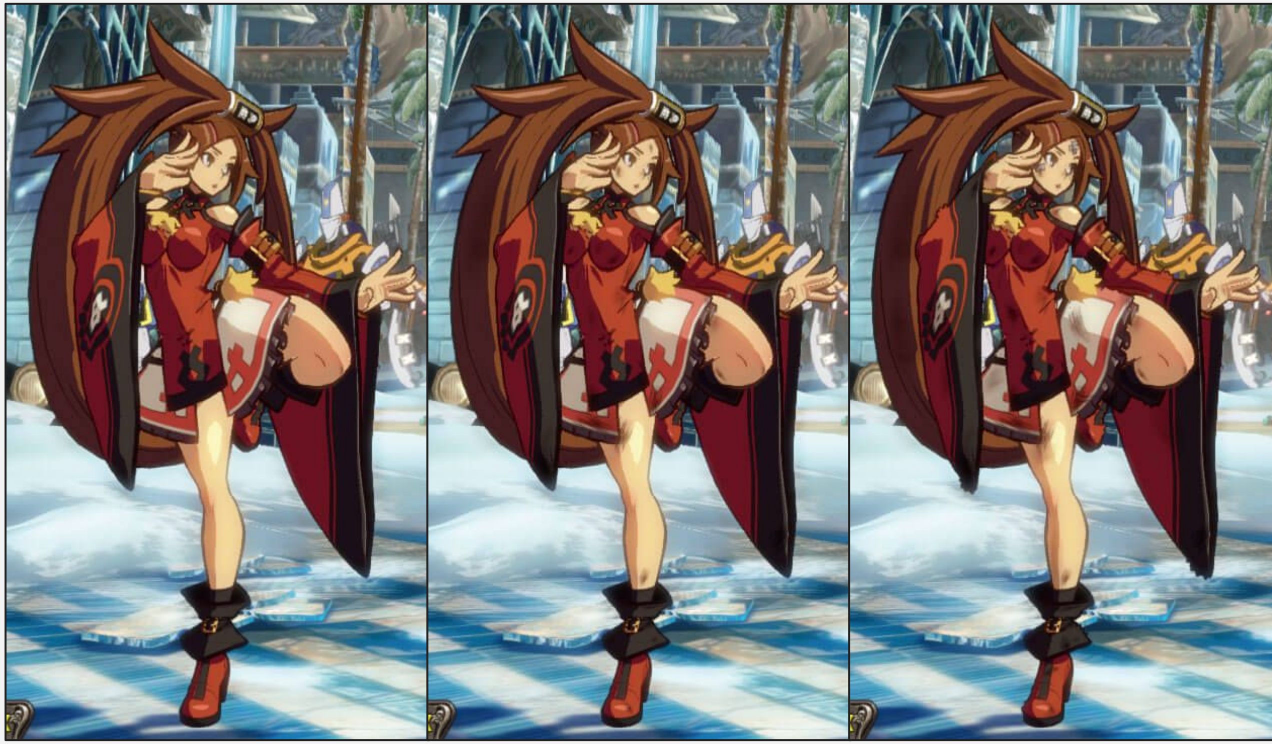

↑ Point Light ON, VFX OFF for clarity


↑ Point Light ON, VFX ON

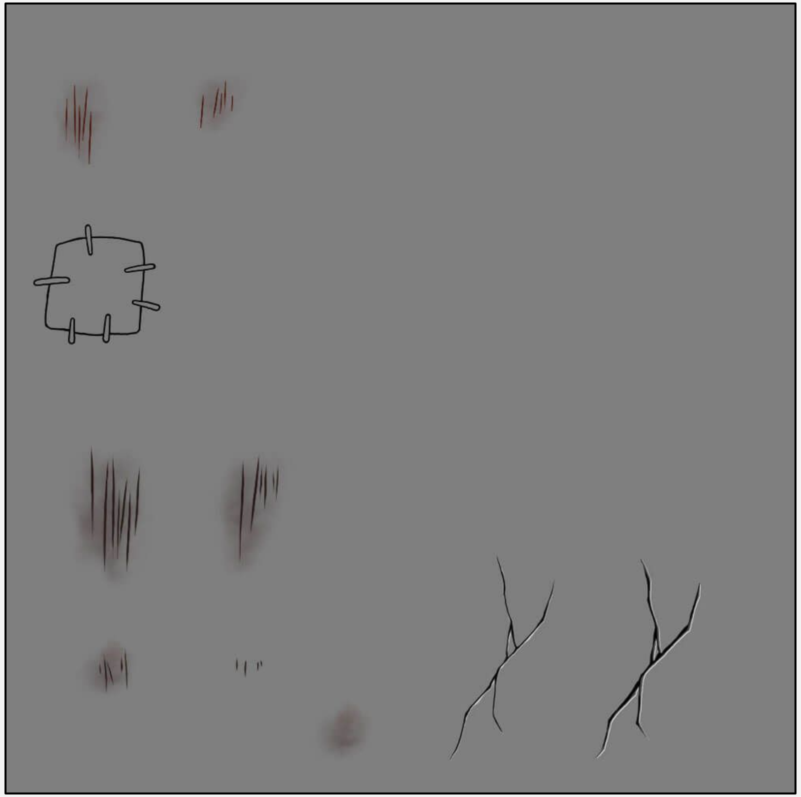
Adding Battle Damage

As an element unique to fighting, a system has been added that applies damage such as scratches to characters in battle. Scratches and stains are shown by applying decal textures on top of the normal state.

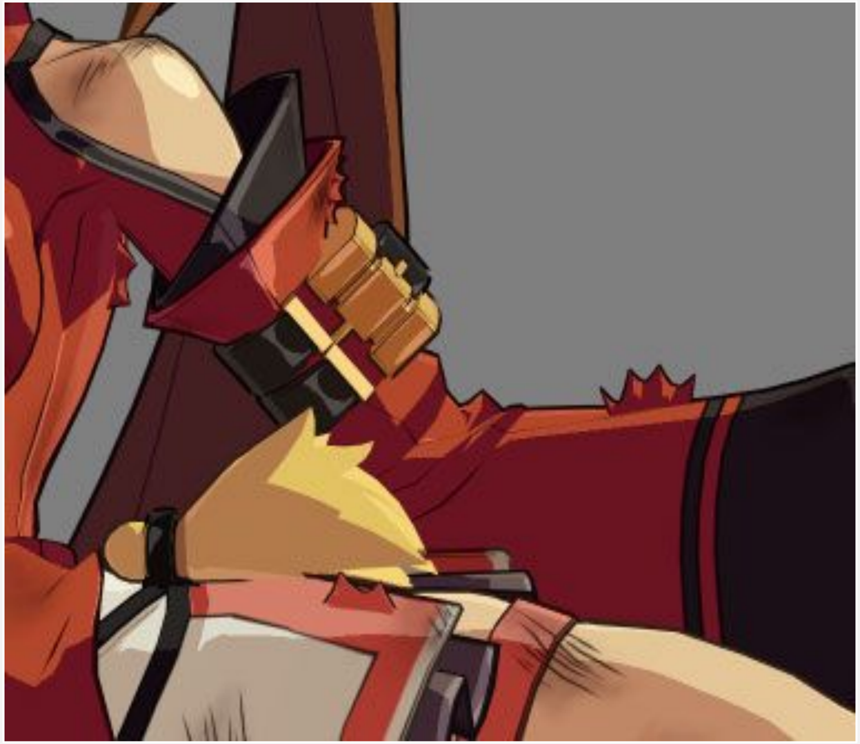
Additionally, torn and ragged meshes were created to show damage to clothing. The look of the shadows is matched to the original via normals. They are finely added to the face because of close-ups, but overall they were added in large sizes to create a change in silhouette and appearance.

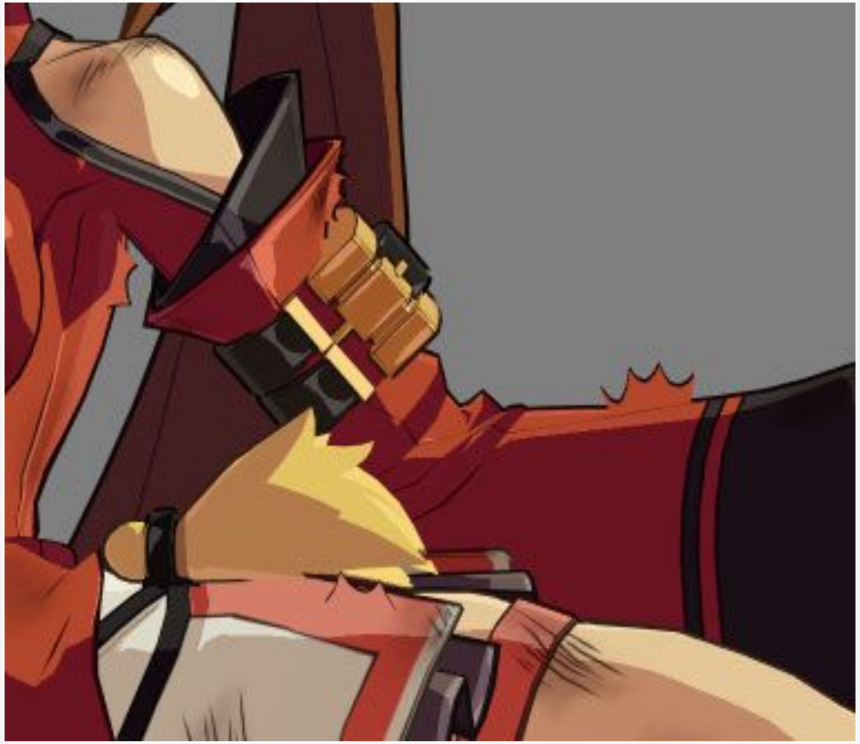


↑ Different stages of battle damage. From the left:
No damage, 1st Level of damage, and final stage of damage.



Decal texture for battle damage.


↑ Torn fabric mesh before editing the normals. The shadows don't match the surface where it's been placed, looks strange.

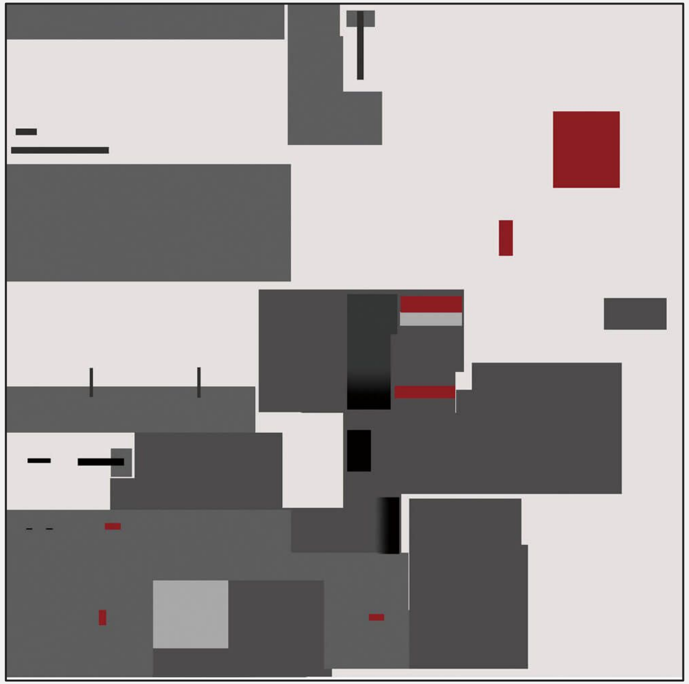

↑ After editing the normals, you can see that the shadows match.

/02/
Anime Stylizations

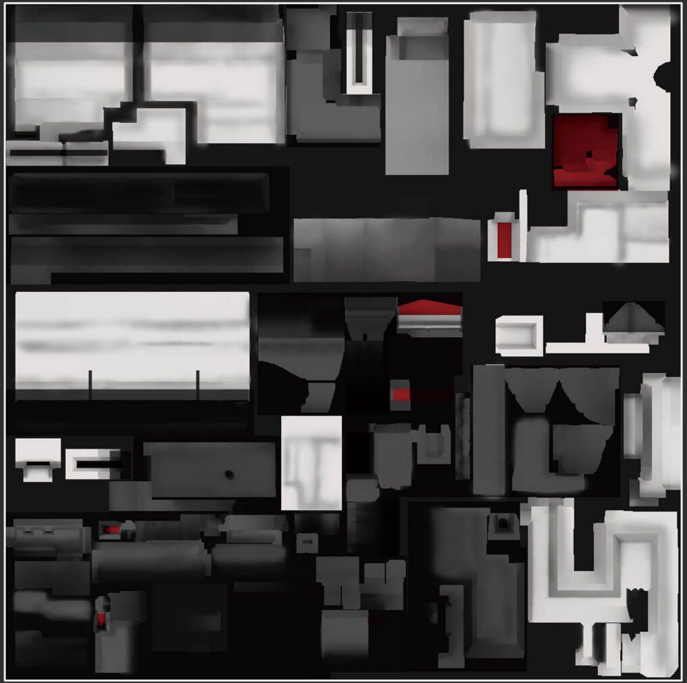
In addition to the “3D Drawing” that we’ve achieved through adding 3D elements to the drawn style, this title adds new innovations such as the metallic surfaces of mecha, to strengthen the anime feeling.

Attention to how Mecha-style metal looks.

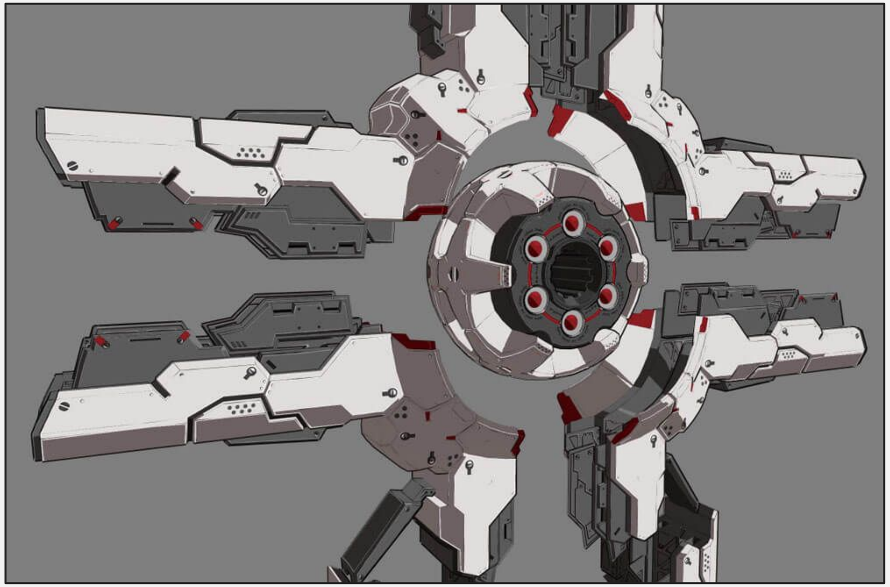
For the newly introduced mechanical textures, multiple methods were used to reproduce an anime-like metal texture. The first is adding ambient occlusion to the texture. Shadows are roughly created using ambient occlusion, and then hand-painted based on those shadows to create the final texture.



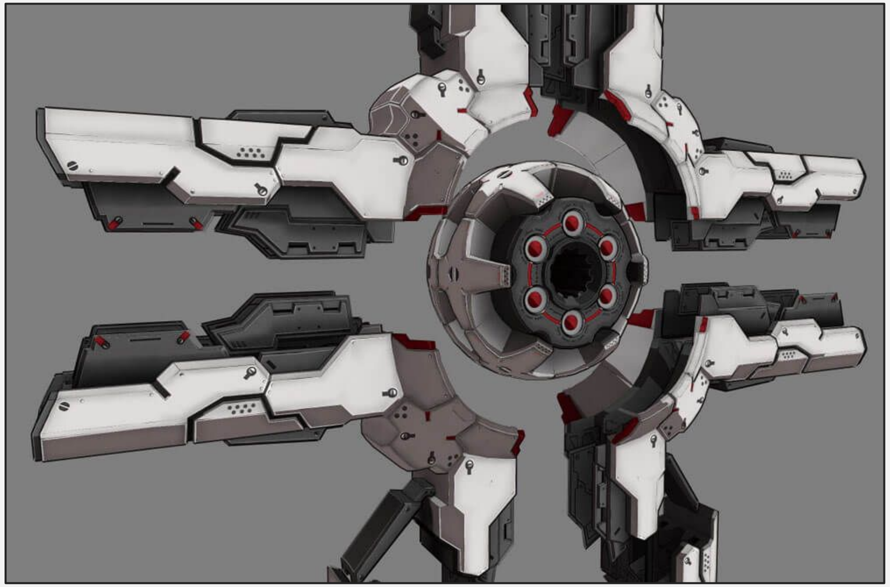
↑ Normal Base Texture



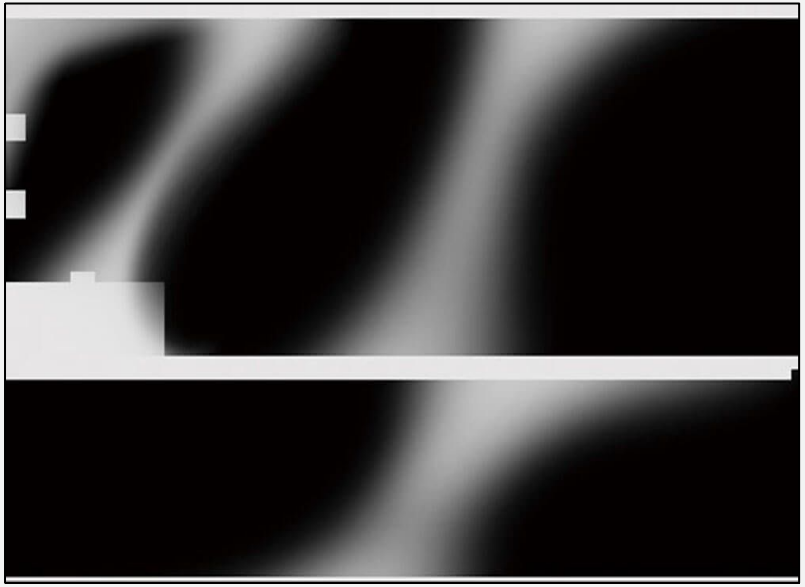
↑ Base Texture with painted AO



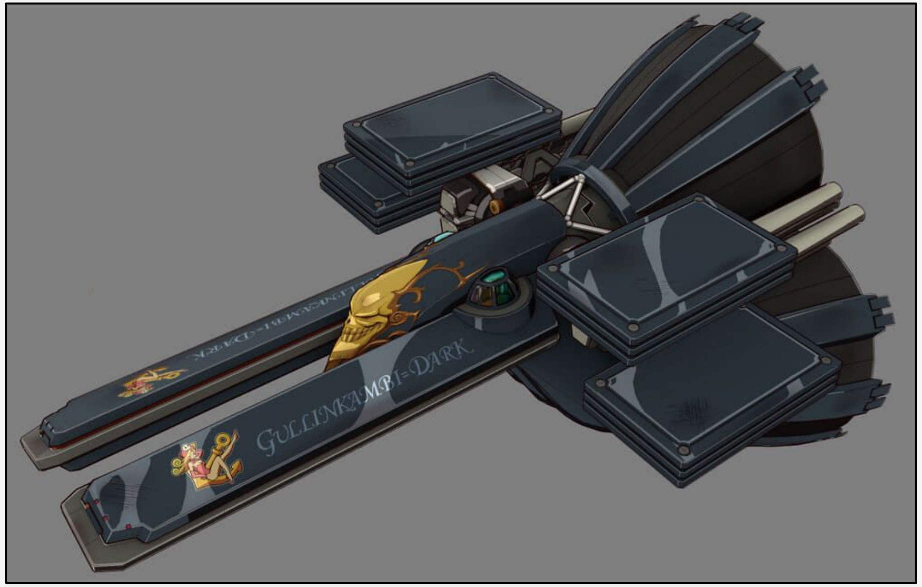
↑ Model with Base Texture



↑ Model with “shaded” texture. You can see how much richer the image is compared to the flat textured model.



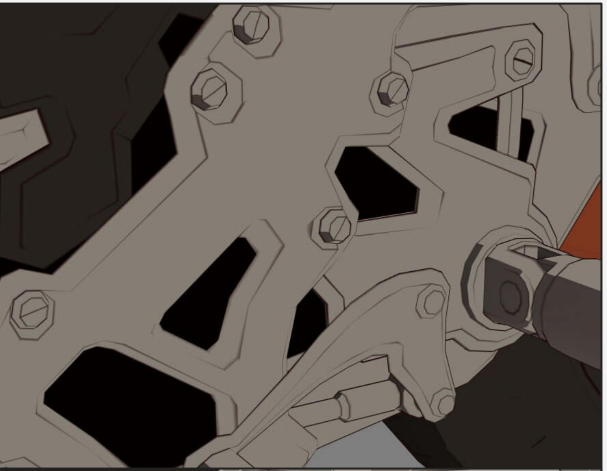
↑ We also implemented a “Wakame Highlight”, which simulates the luster of metal in an anime-style.[1]



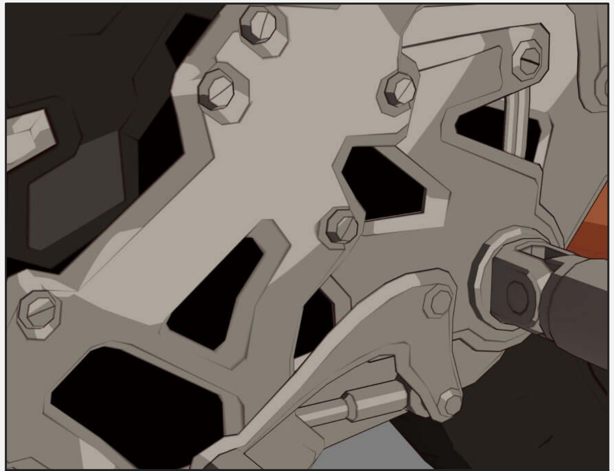
↑ Model with the Wakame Highlight texture applied. The texture is designed to look like the seaweed, and is adjusted to produce the desired shape of highlights.

TN[1]: Wakame-style shadows (ワカメ影) are a popular technique in mecha anime of the 1980s, popularized by Masahito Yamashita. Sakugabooru entry.

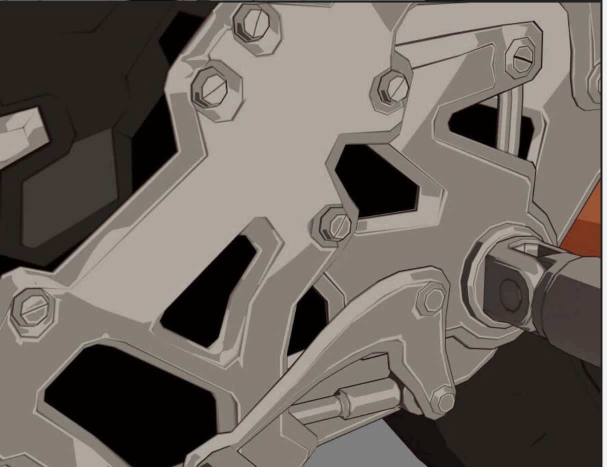
Special highlights have also been added to edges, to enhance the metallic material.



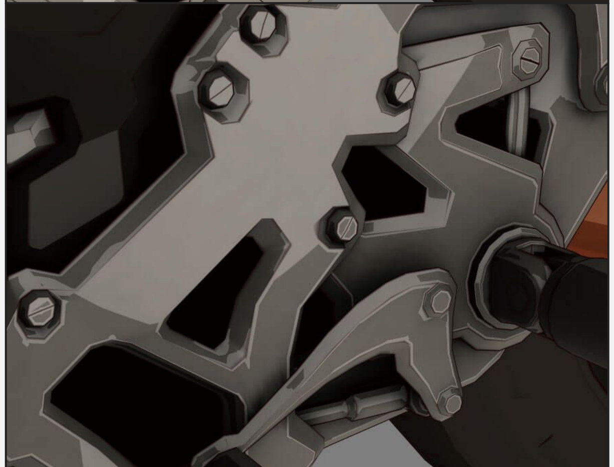
← Color Only



← Added Highlights



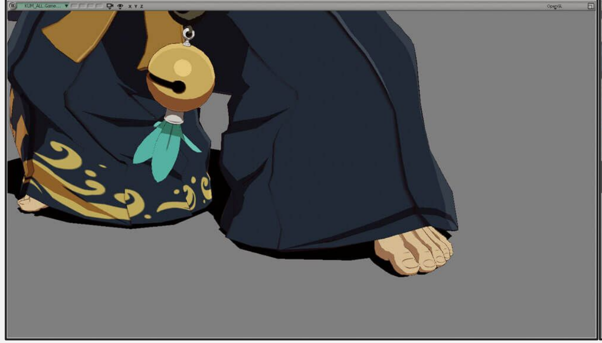
← With Highlights and
Edge Highlights



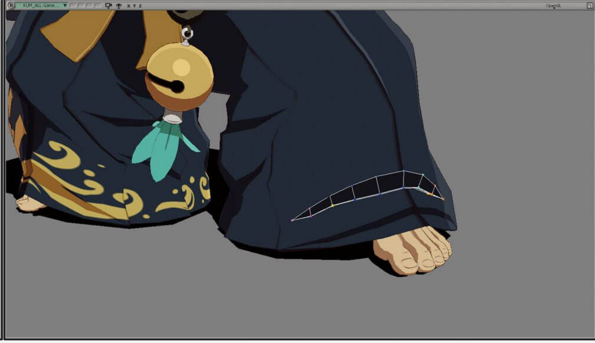
← Everything together with the painted texture.

Drawing with 3D

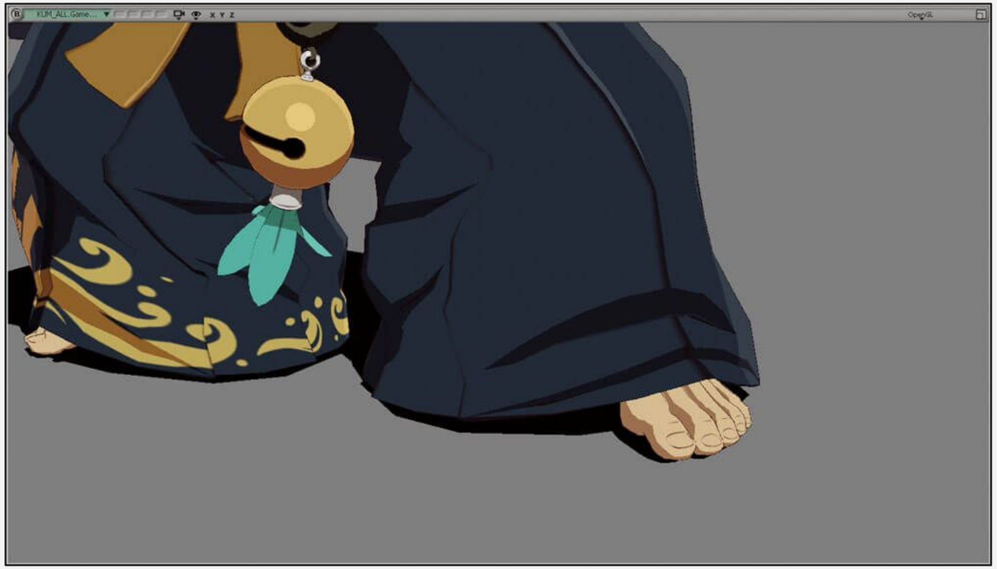
In the Xrd series, the animation workflow includes traditional drawing, so it was important to control the amount of information in a single frame, for example, to express a specific facial expression or shadow shape. Because of this, we prepared many unique facial expressions and extra parts, and the animators them in 3D space as if they were drawing. *“It's a manual method, but by preparing many small pieces, we were able to achieve the image we wanted"* says Sakamura.



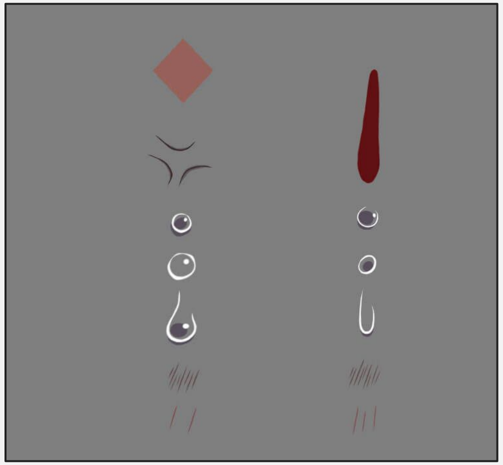
↑ Normal



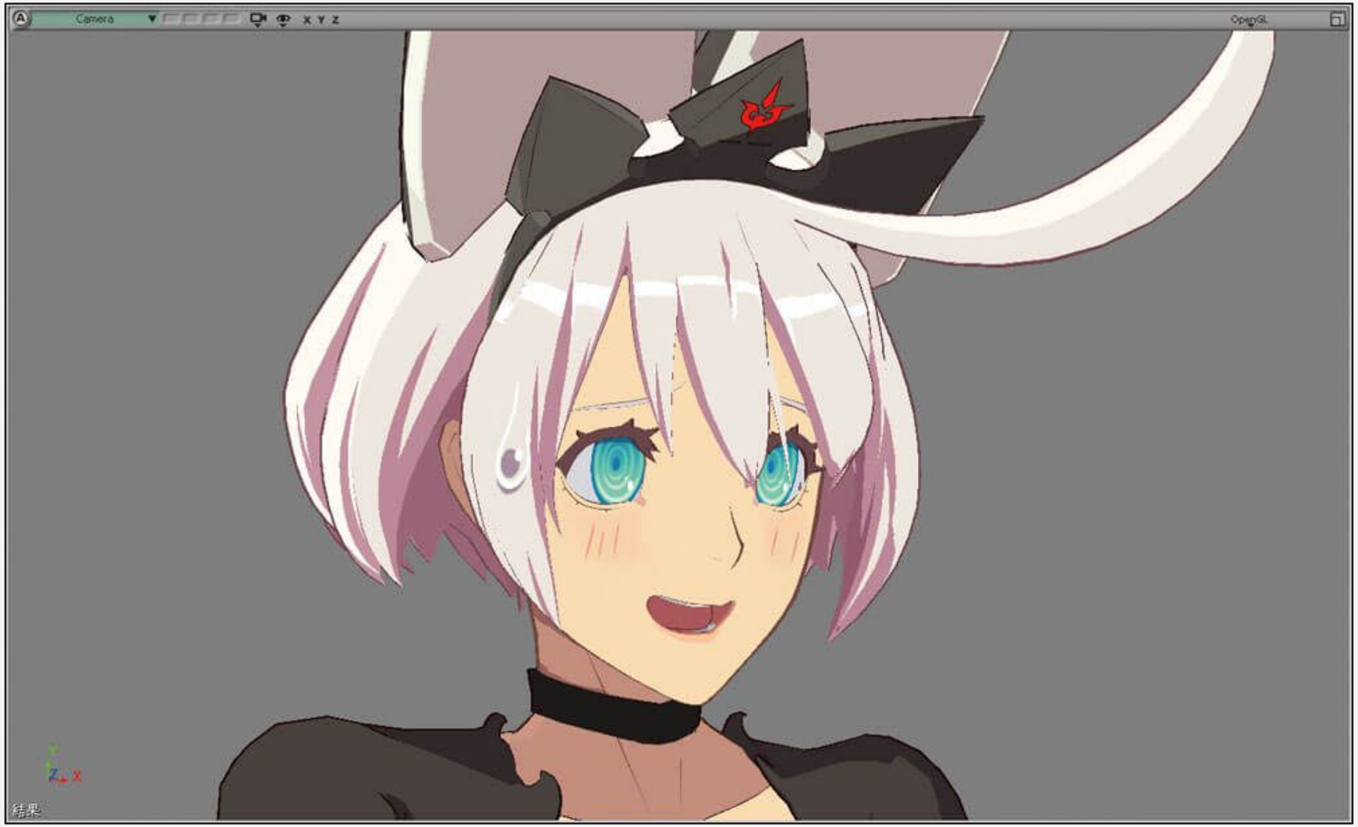
↑ Adding topology where we want a shadow



↑ Shadow ON



← Some expression parts



Placed on the character →

/03/
Battle Stages

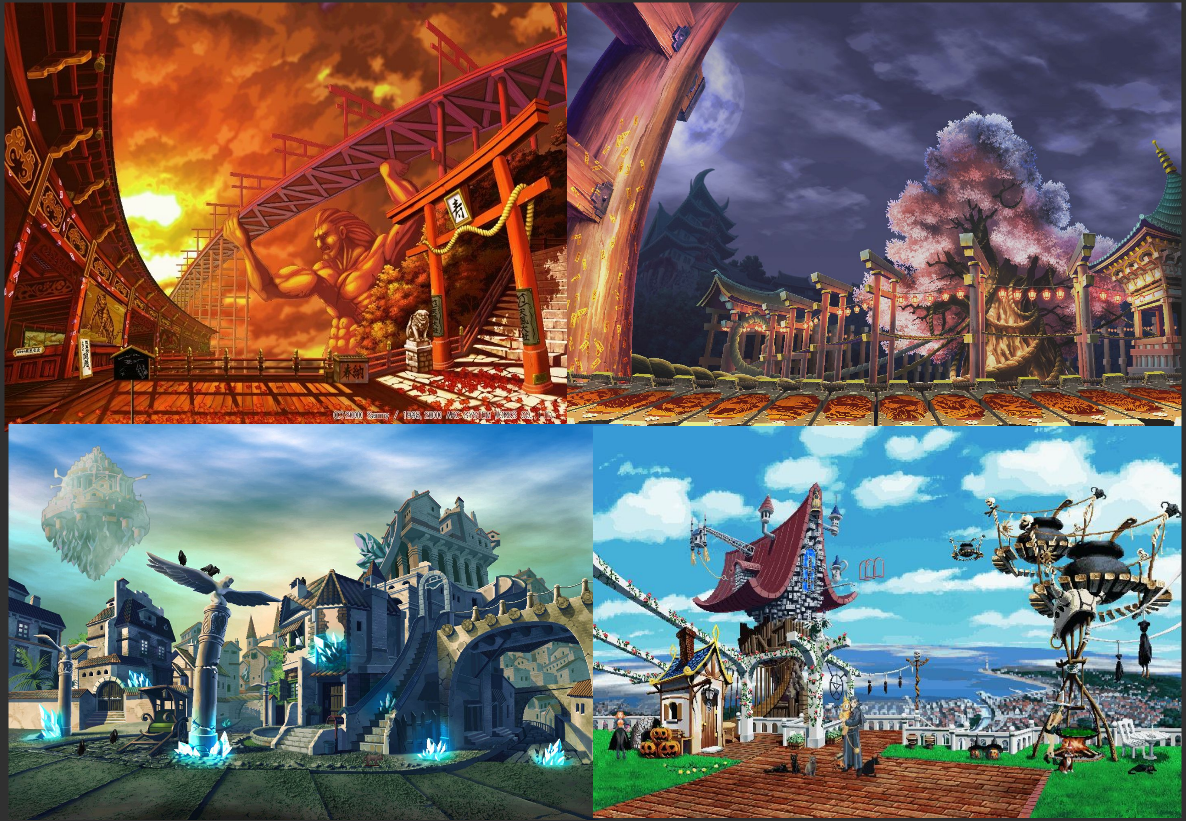
In keeping with the theory of new fighting games, the backgrounds have undergone a major renovation and refinement. When the camera rolls in and out, a lot of thought was put into making the stage look good.

Creating stages for Fighting Games

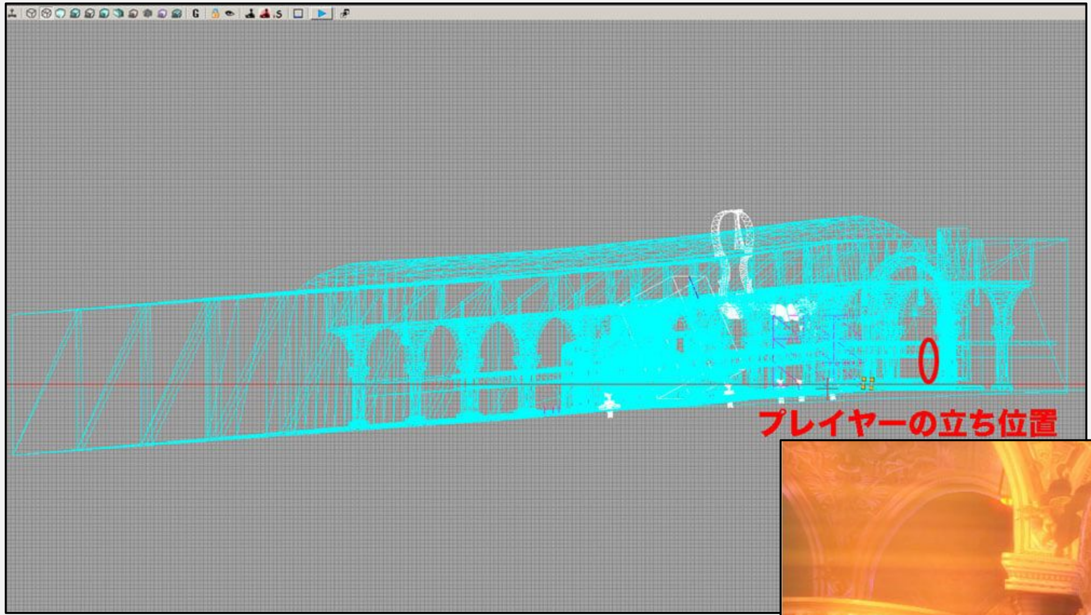
There are certain rules and formats that have been built up over time for backgrounds in fighting games. Extreme exaggerations using fisheye lenses and false perspectives.

In the background of fighting games, there are certain rules and formats that have been built up over the years. For example, extreme exaggeration using fisheye lenses and false perspective, information density, a sense of “layering” through intentional scrolling of background, middle, and foreground elements, and individual designs for the center, right, and left edges of the screen.

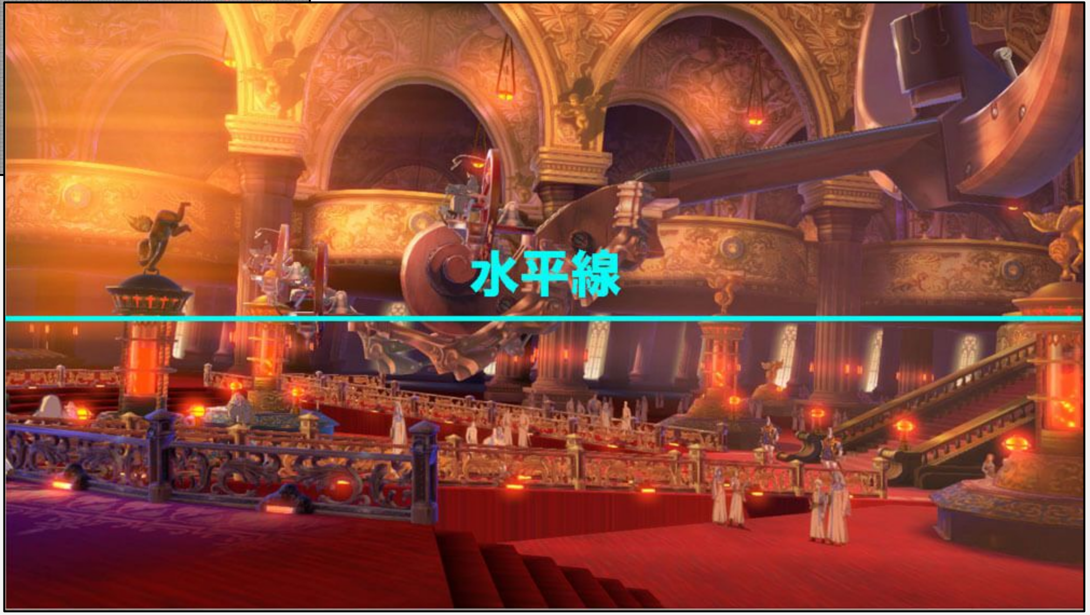
The battle stages in this game were created in accordance with these rules.



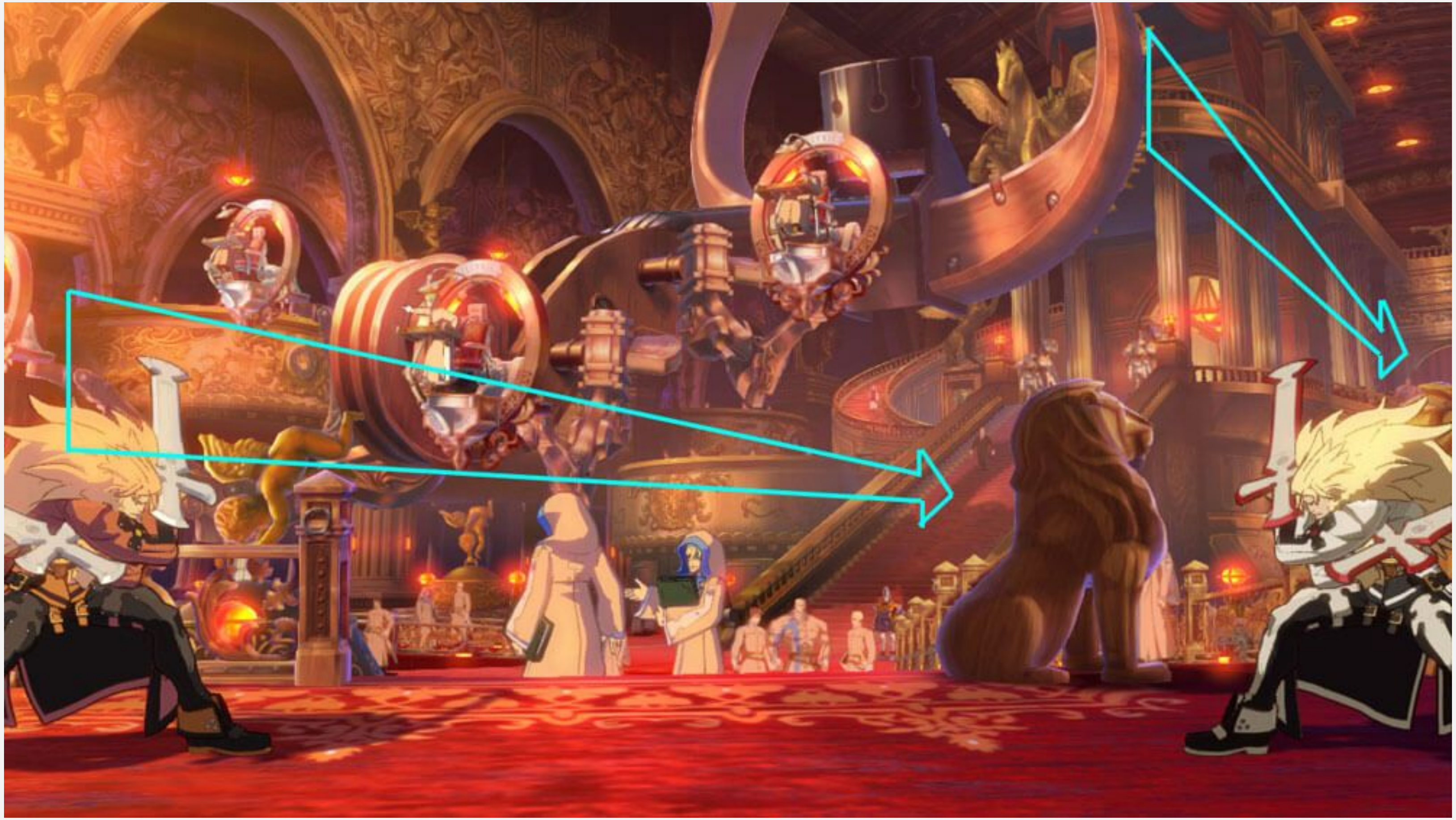
↑ Classic Guilty Gear Stages



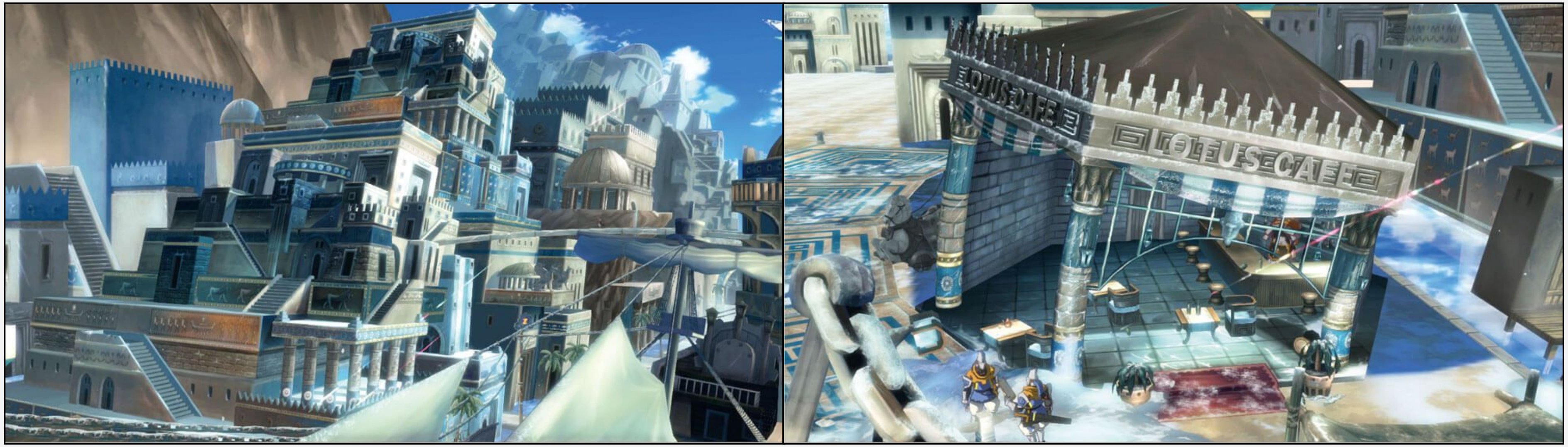
← Side view of the “War Room” Stage



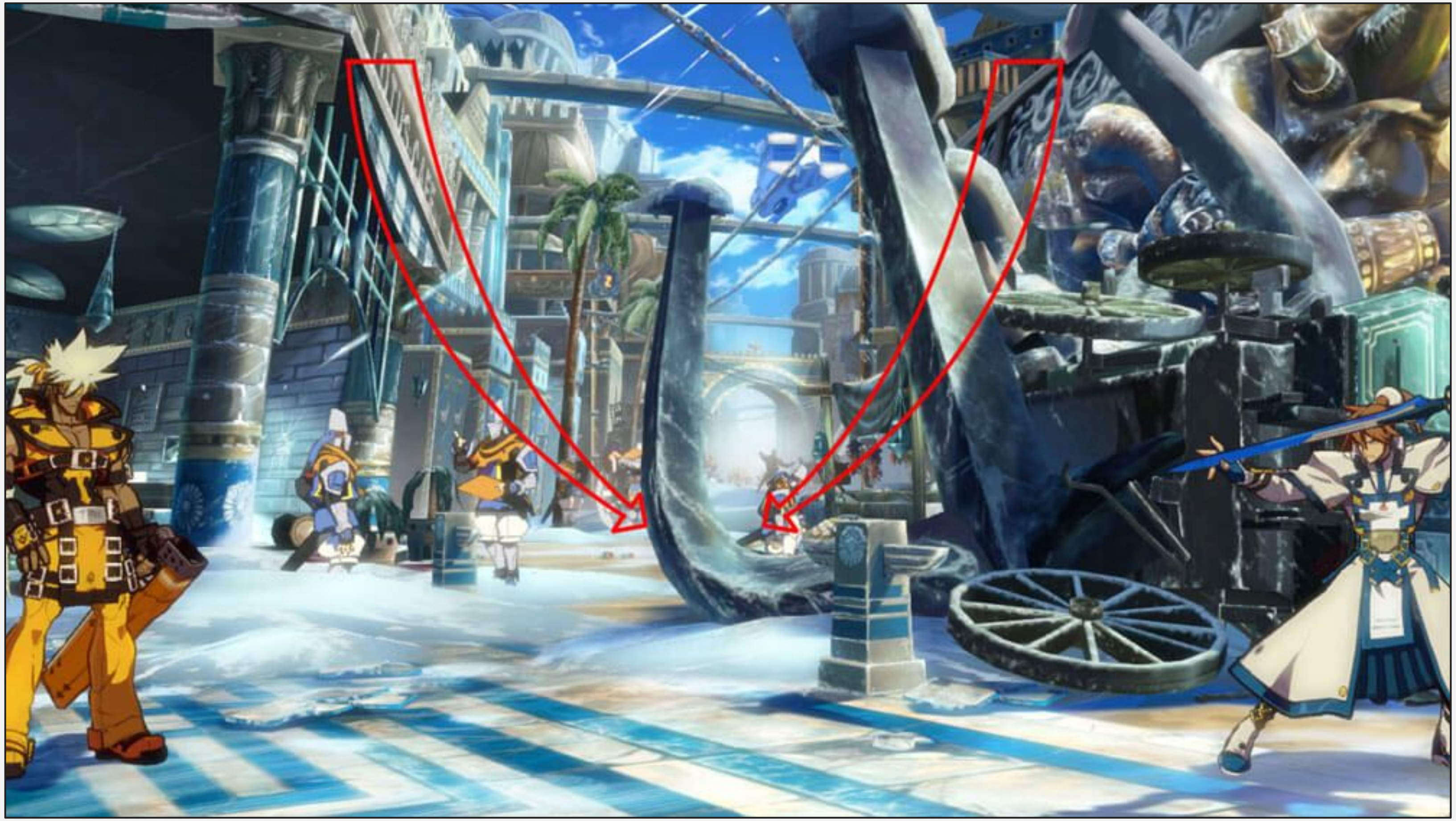
Another Angle.
The image shows the horizon line →



↑ The actual screen. By lowering the horizon diagonally from the player character to the back of the screen, we've created a stage with perspective and depth.



↑ “Babylon” stage, as seen from different angles.



↑ The actual screen. The mesh is bent to create an exaggerated look, like a super wide-angle lens.

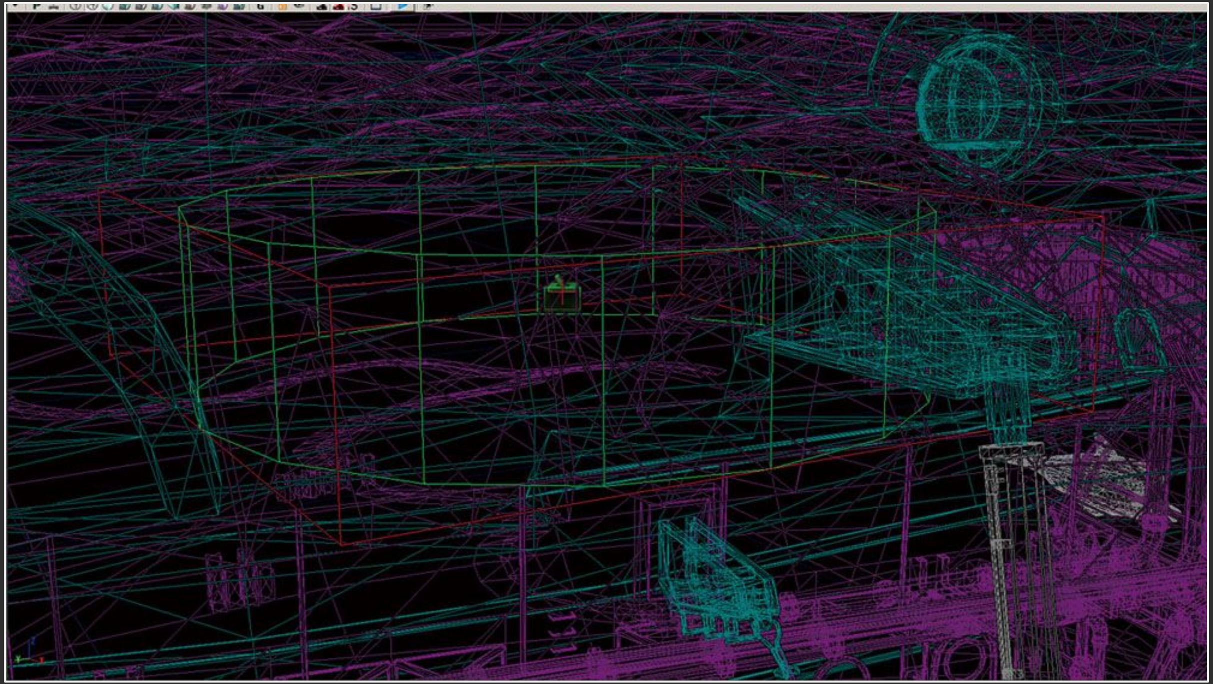
Supporting cutscene cameras

When the camera is directed to move, like during a special attack, there are inevitably parts in the stage that would get in the way of the shot.

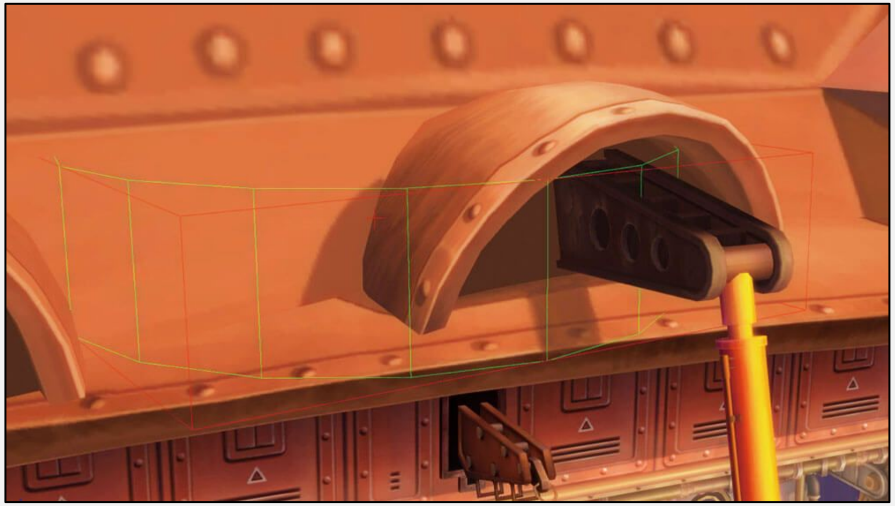
In order to avoid this, we’ve introduced a system to cull parts.

The hiding and showing of parts is controlled by UE3's Kismet.[1]

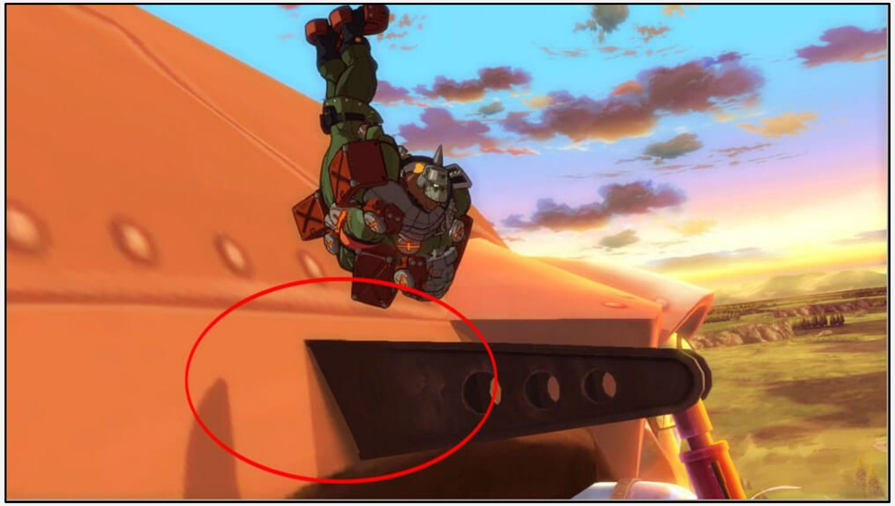
TN[1]: Kismet is UE3’s ancestor equivalent to UE4 and 5’s Blueprint system.



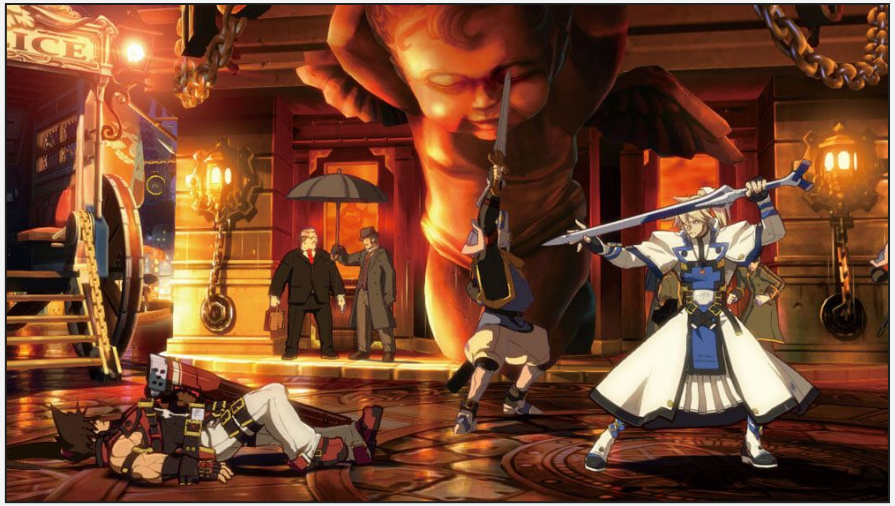
↑ Stage’s culling area. Whenever the camera enters this space, specific objects will be hidden.



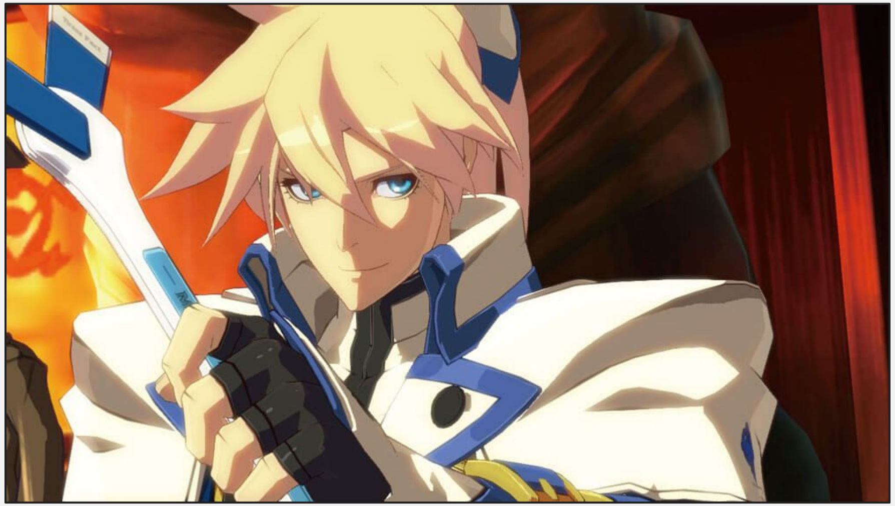
↑ Hiding range and object that will be hidden.



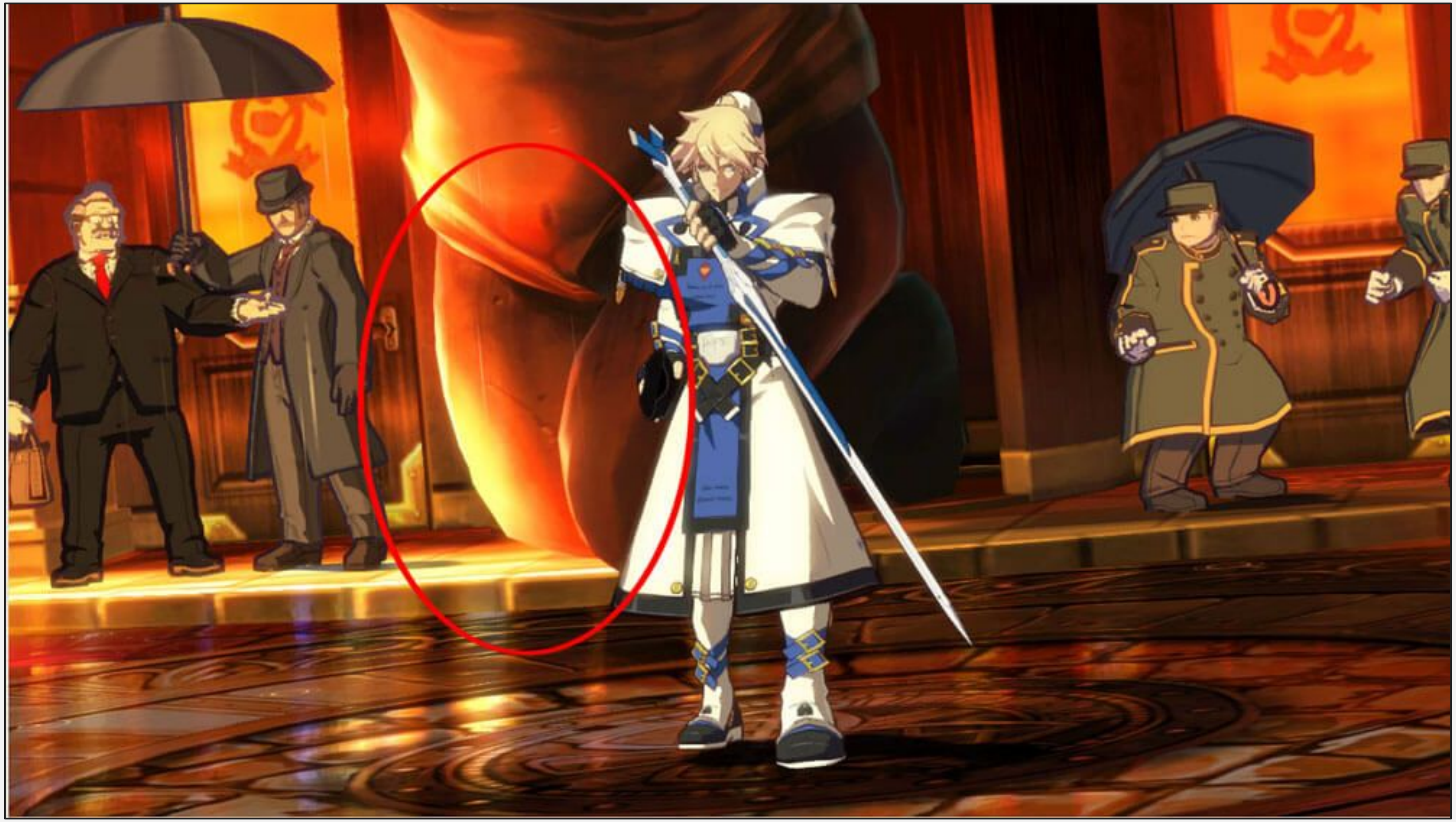
↑ A different angle of the cutscene. You can see the objects from the previous image are hidden.



↑ Not just background objects, but mob characters can be hidden as well. Here’s a normal view, with a mob in the center.



↑ Cutscene camera. The mob character has disappeared.



↑ Different angle of the cutscene. You can see that the mob character is gone.

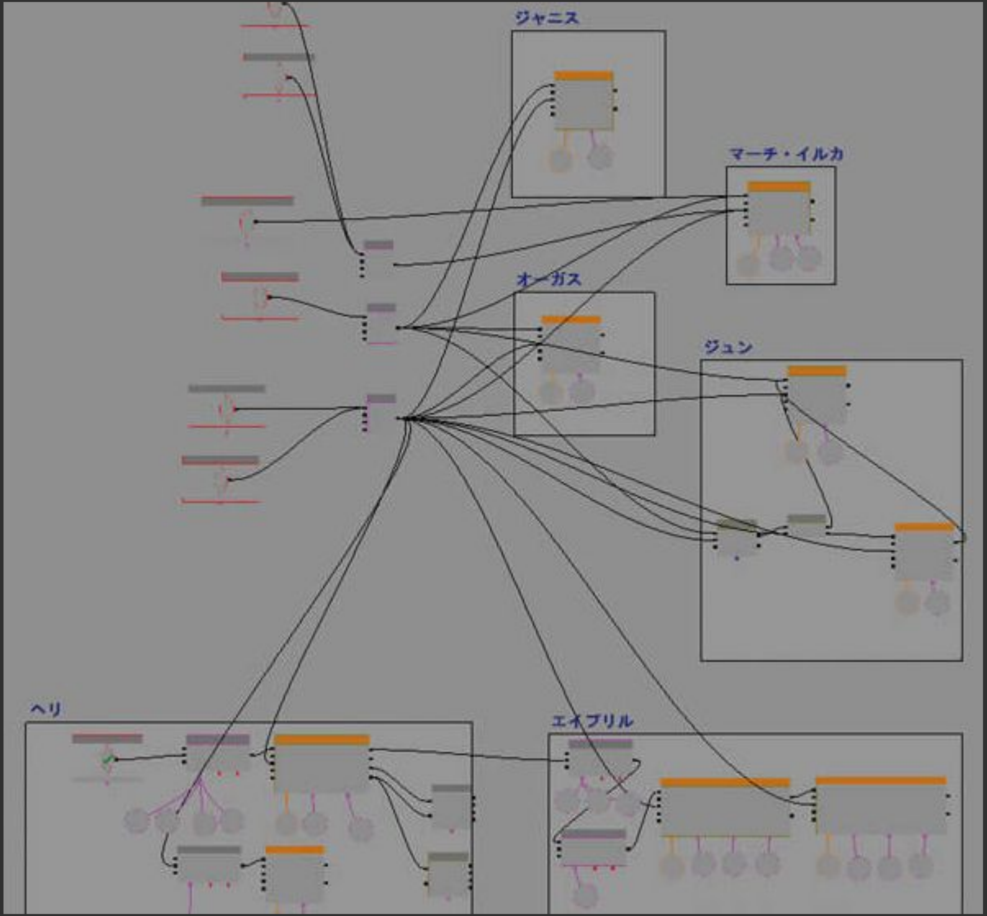
/04/
Battle Stages 2

In this title, the mob characters and gimmicks in the stages change as time passes and the rounds progress. This is a fun feature that will keep you coming back for more.

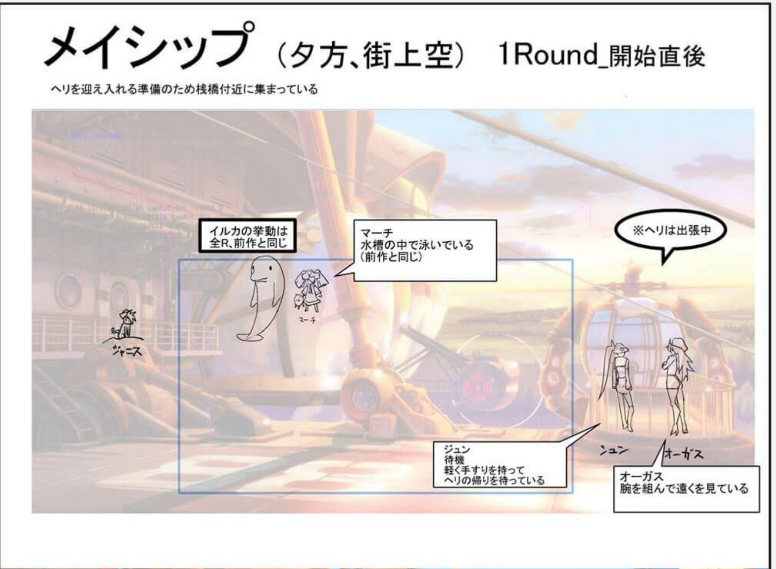
Mobs and Stage Gimmicks

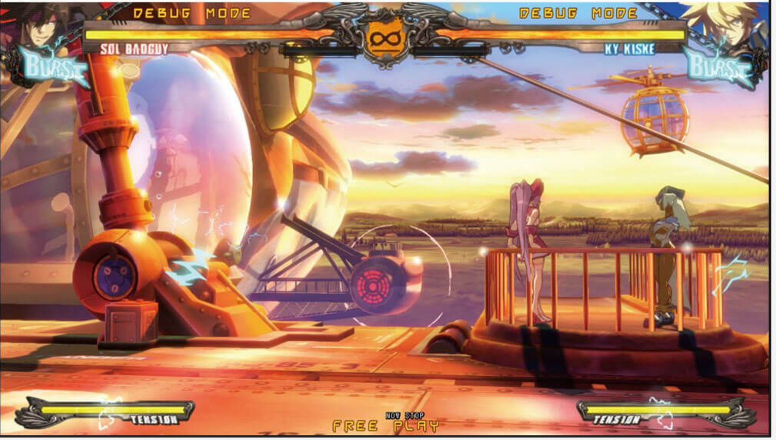
The backgrounds are designed to change with as time passes and you go through the rounds. First, the background team designs the production plan in 2D, then the animation team adds movement to it, and then the background team actually incorporates it into the game.

This is also controlled by Kismet, and is triggered by each sequence called "round phase". This includes the passage of time, weather changes, victory or defeat, and the display or non-display of objects.



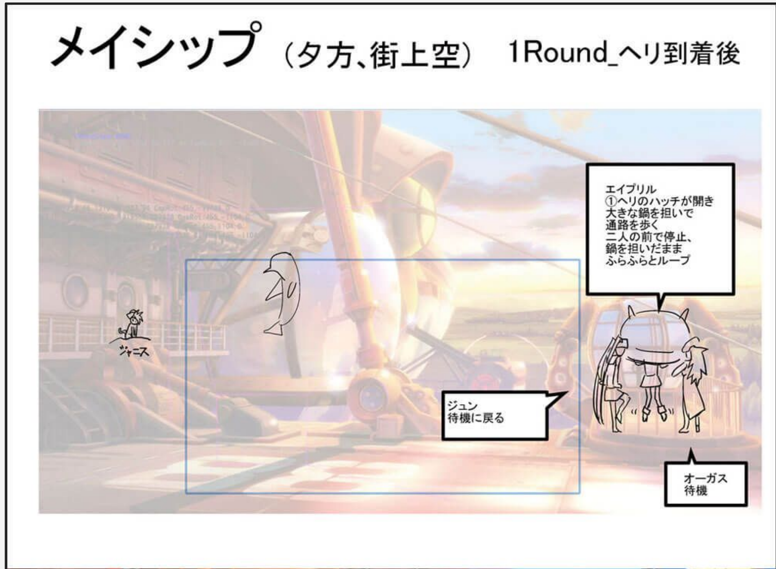
↑ Kismet in Round 1

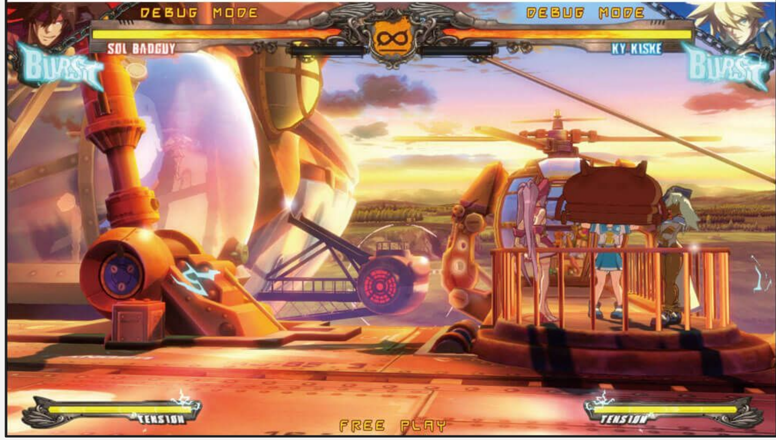

メイシップ（夕方、街上空） 1Round_開始直後
ヘリを迎え入れる準備のため桟橋付近に集まっている
イルカの挙動は全R、前作と同じ
マーチ 水槽の中で泳いでいる（前作と同じ）
※ヘリは出張中
マーチ
ジャニス
ジュン 待機 軽く手すりを持って ヘリの帰りを待っている
ジュン
オーガス
オーガス 腕を組んで遠くを見ている

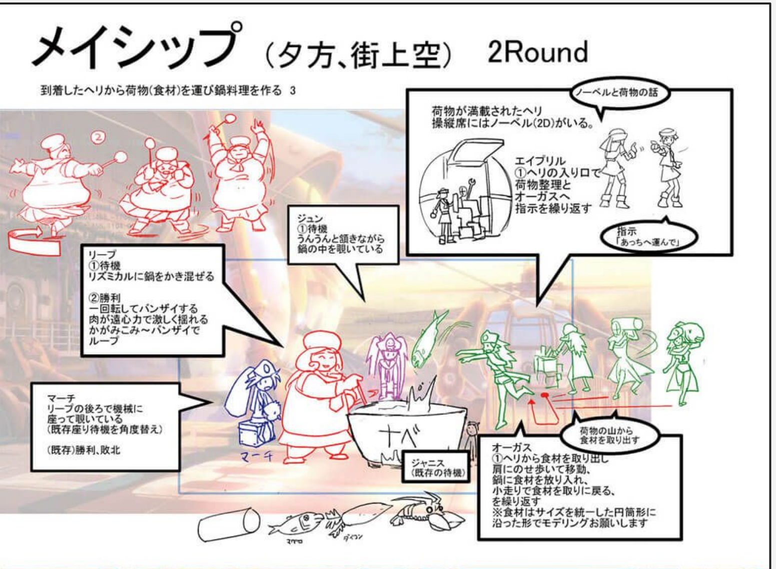


← The production plan and screenshot at the start of round. Waiting for the helicopter to arrive.

Stage plan and screenshot after the heli arrives in round 1. A girl carrying a large pot walks in. →


メイシップ（夕方、街上空） 1Round_ヘリ到着後
エイプリル ①ヘリのハッチが開き 大きな鍋を担いで 通路を歩く 二人の前で停止、 鍋を担いだまま ふらふらとループ
ジャニス
ジュン 待機に戻る
オーガス 待機




メイシップ （夕方、街上空） 2Round

到着したヘリから荷物(食材)を運び鍋料理を作る 3

ノーベルと荷物の話

荷物が満載されたヘリ
操縦席にはノーベル(2D)がいる。

エイプリル
①ヘリの入り口で
荷物整理と
オーガスへ
指示を繰り返す

指示
「あっちへ運んで」

ジュン
①待機
うんうんと頷きながら
鍋の中を覗いている

リープ
①待機
リズミカルに鍋をかき混ぜる

②勝利
一回転してバンザイする
肉が遠心力で激しく揺れる
かがみこみ～バンザイで
ループ

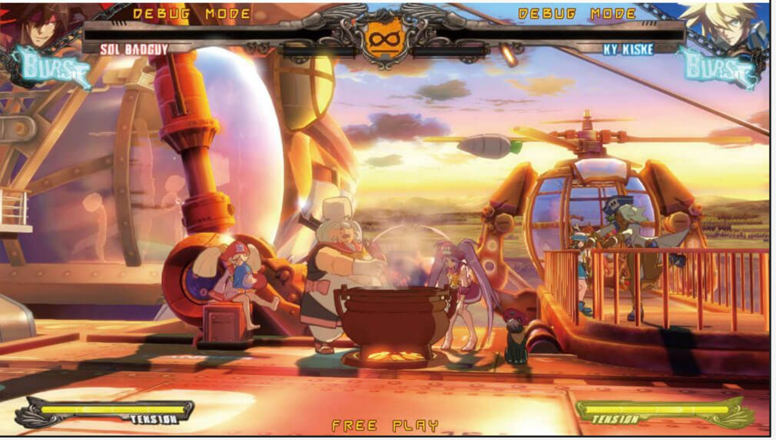
マーチ
リープの後ろで機械に
座って覗いている
(既存座り待機を角度替え)

(既存)勝利、敗北

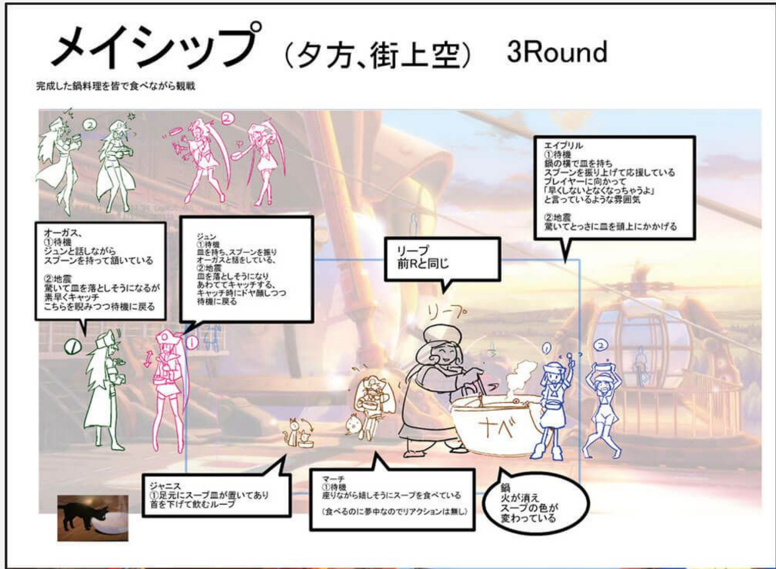
ジャニス
(既存の待機)

荷物の山から
食材を取り出す

オーガス
①ヘリから食材を取り出し
肩にのせ歩いて移動、
鍋に食材を放り入れ、
小走りで食材を取りに戻る、
を繰り返す
※食材はサイズを統一した円筒形に
沿った形でモデリングお願いします



← Round 2 plan and screenshot. Throw the ingredients in the pot and cooking.


メイシップ （夕方、街上空） 3Round

完成した鍋料理を皆で食べながら観戦

オーガス、
①待機
ジュンと話しながら
スプーンを持って頷いている

②地震
驚いて皿を落としそうになるが
素早くキャッチ
こちらを睨みつつ待機に戻る

ジュン
①待機
皿を持ち、スプーンを振り
オーガスと話をしている、
②地震
皿を落としそうになり
あわててキャッチする、
キャッチ時にドヤ顔しつつ
待機に戻る

リープ
前Rと同じ

エイプリル
①待機
鍋の横で皿を持ち
スプーンを振り上げて応援している
プレイヤーに向かって
「早くしないとなくなっちゃうよ」
と言っているような雰囲気

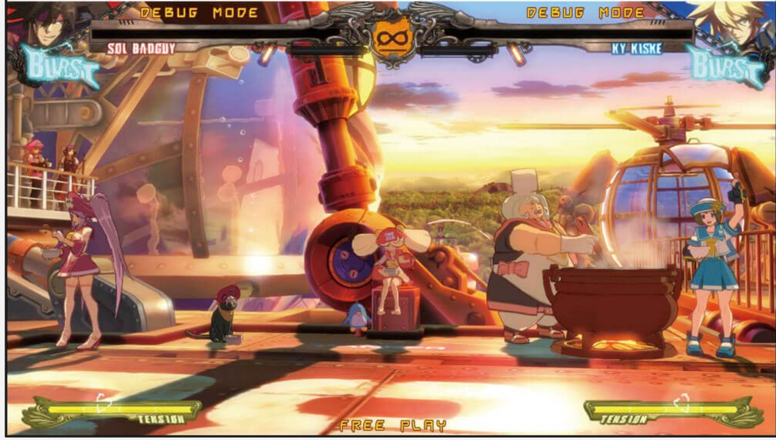
②地震
驚いてとっさに皿を頭上にかかげる

ジャニス
①足元にスープ皿が置いてあり
首を下げて飲むループ

マーチ
①待機
座りながら嬉しそうにスープを食べている

(食べるのに夢中なのでリアクションは無し)

鍋
火が消え
スープの色が
変わっている



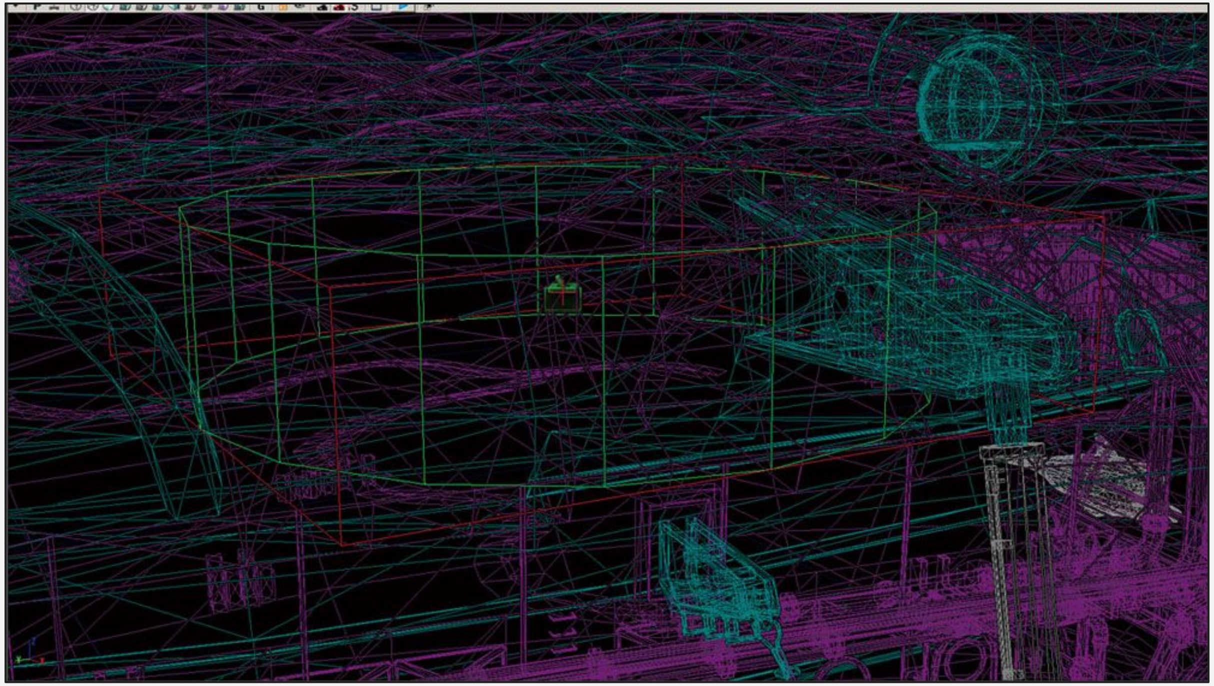
Plan for round 3 and screenshot. The characters are watching the battle while eating! →

Supporting cutscene cameras

When the camera is directed to move, like during a special attack, there are inevitably parts in the stage that would get in the way of the shot.

In order to avoid this, we’ve introduced a system to cull parts.

The hiding and showing of parts is controlled by UE3's Kismet.¹



↑ Stage’s culling area. Whenever the camera enters this space, specific objects will be hidden.

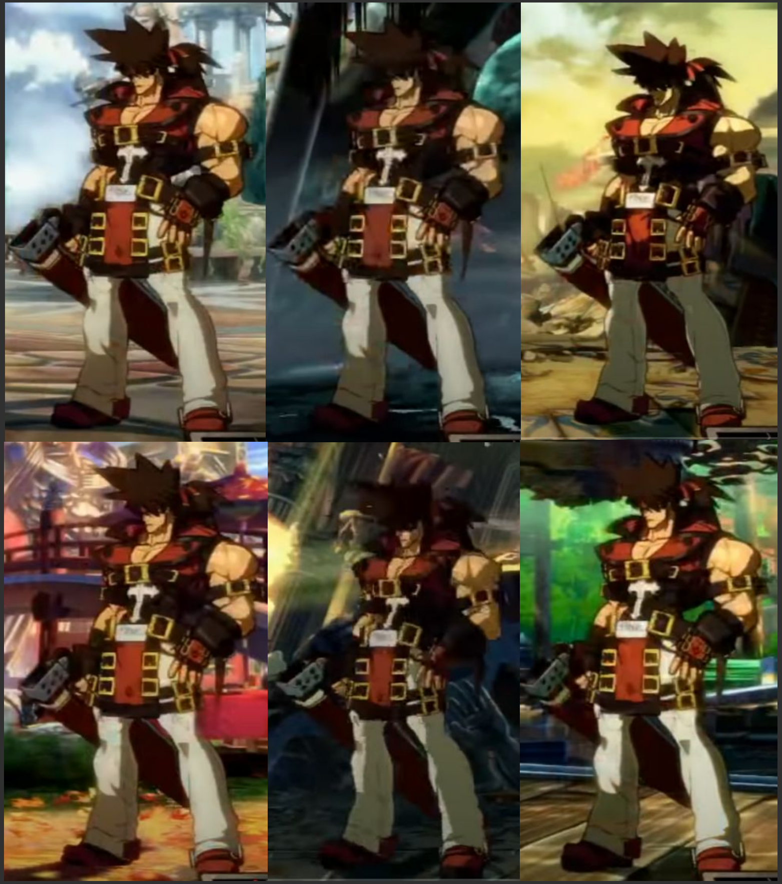
TN¹: Kismet is UE3’s ancestor equivalent to UE4 and 5’s Blueprint system.

Placing point lights and scene colors

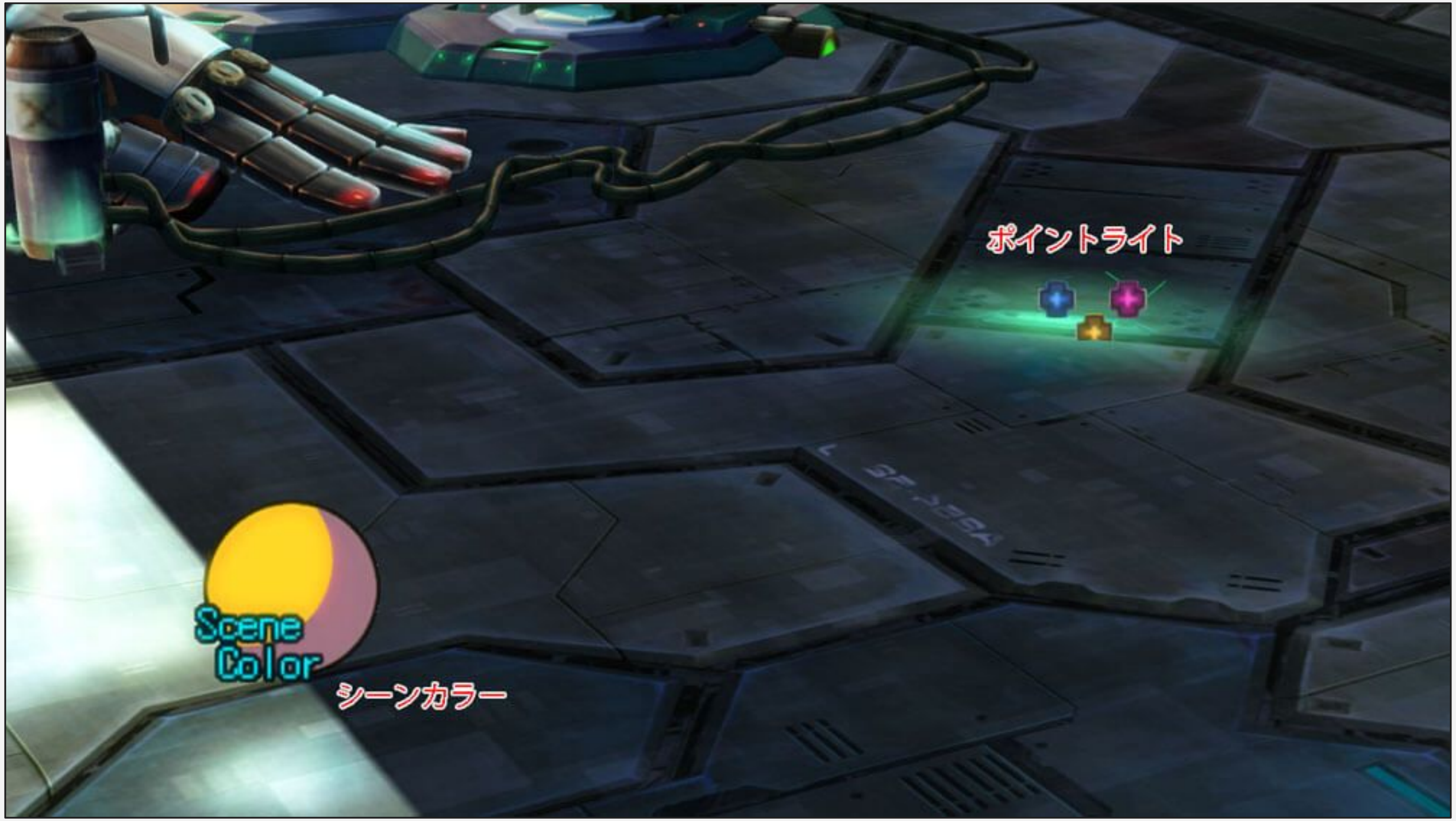
As mentioned before, if there are light sources in the stage, the character is affected by those point lights.

Not only does the color change, but the contrast is also controlled, and the darker look while in shadows is also due to point lights.

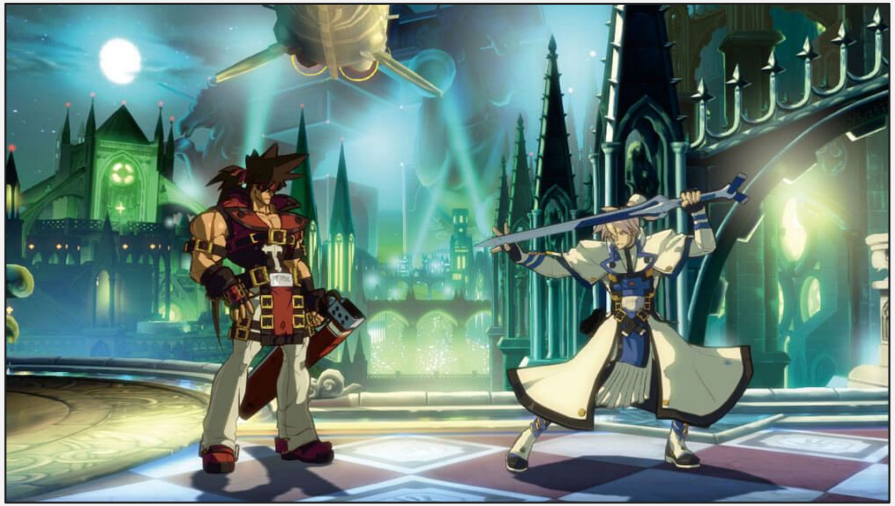
Furthermore, each stage is designed with a separate color for each character. In keeping with the competitive nature of fighting games, and to ensure visibility, a full-time staff member is assigned to manage this very carefully.



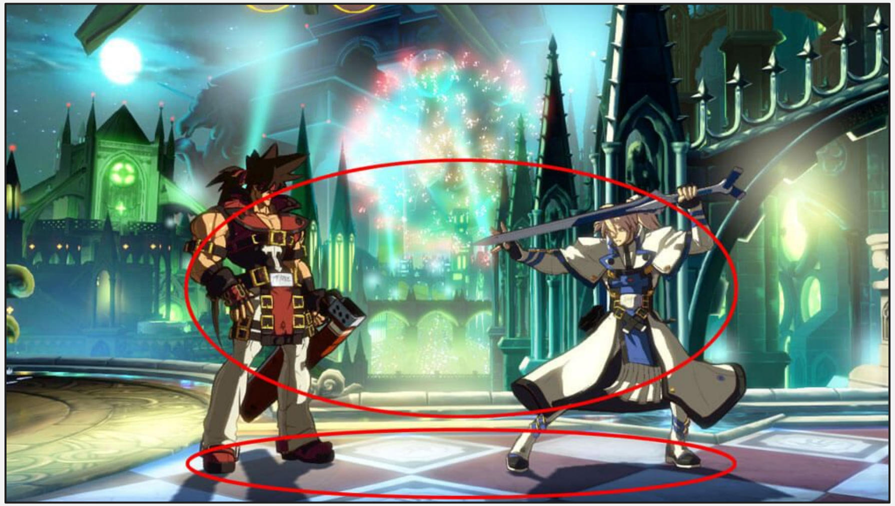
↑ Sol under different lighting conditions



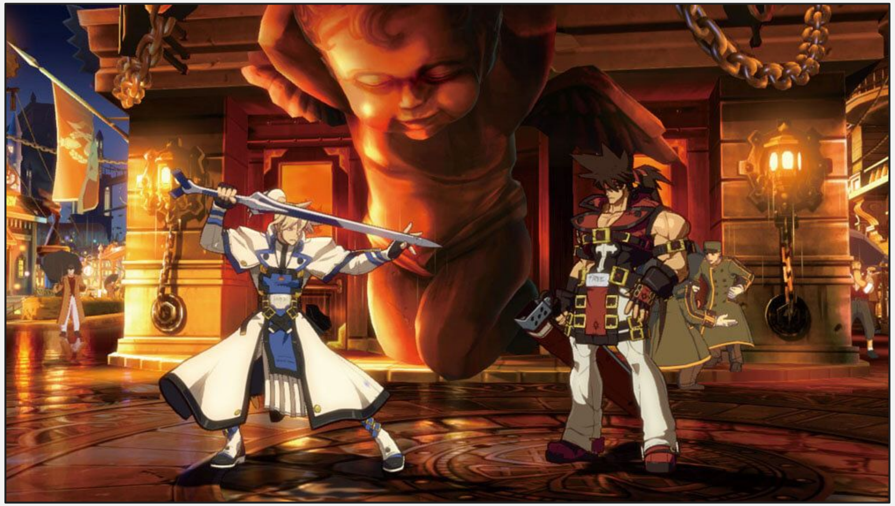
↑ The sphere in the foreground is the scene color. The light source in the back is the point light. These are placed as actors within the UE3 scene.



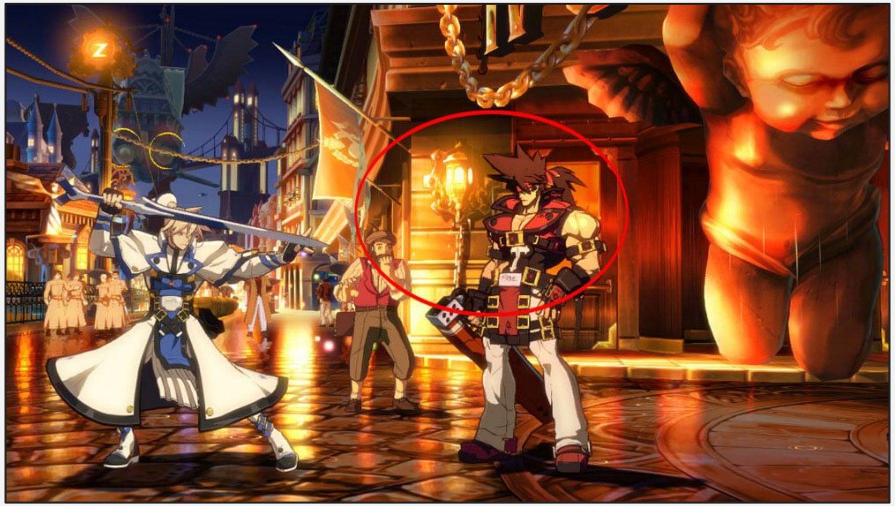
↑ Normal lighting.



↑ Affected by the pointlight. When the fireworks go off, the shadows of the characters become darker, and their cast shadow is stretched.



↑ Normal lighting.



↑ Affected by the pointlight. You can see the effect of the lighting fixtures.

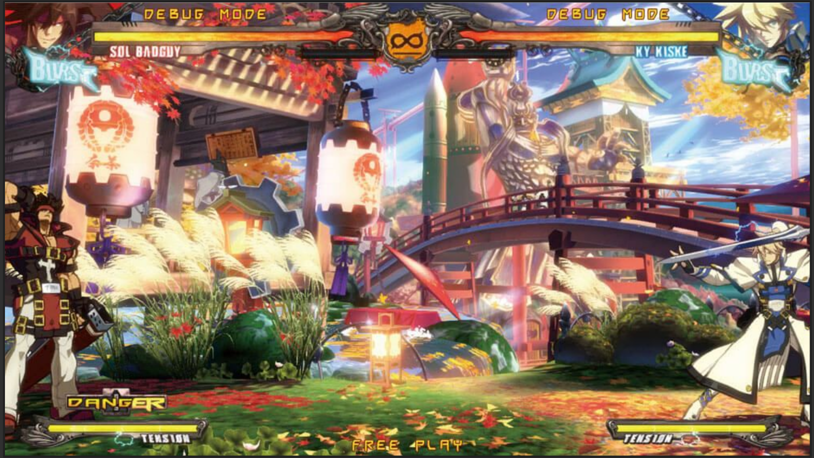
/05/
Story Mode Stages

Since the story backgrounds, which used to be mainly 2D, are almost entirely 3D this time around, we created a workflow to reduce man-hours, and by making full use of UE3's functions, we were able to create something rich, very quickly.

Reusing assets

Since the backgrounds in story mode are now 3D-centric, the assets created for the battle stage were processed and used.

The angle of view and perspective were changed to make it look better, and UE3's umap was used to create a map that makes extensive use of asset placement.

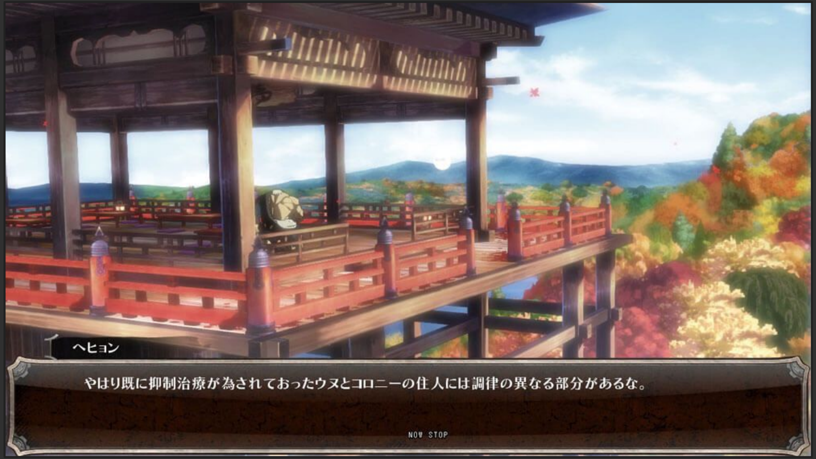


← The battle stage.

This background was created with a 360º view in mind, for use as a location in the Story mode stage.

The story stage.

The hut and mountains in the distance were placed in exact proportions of the models in the battle stage. →



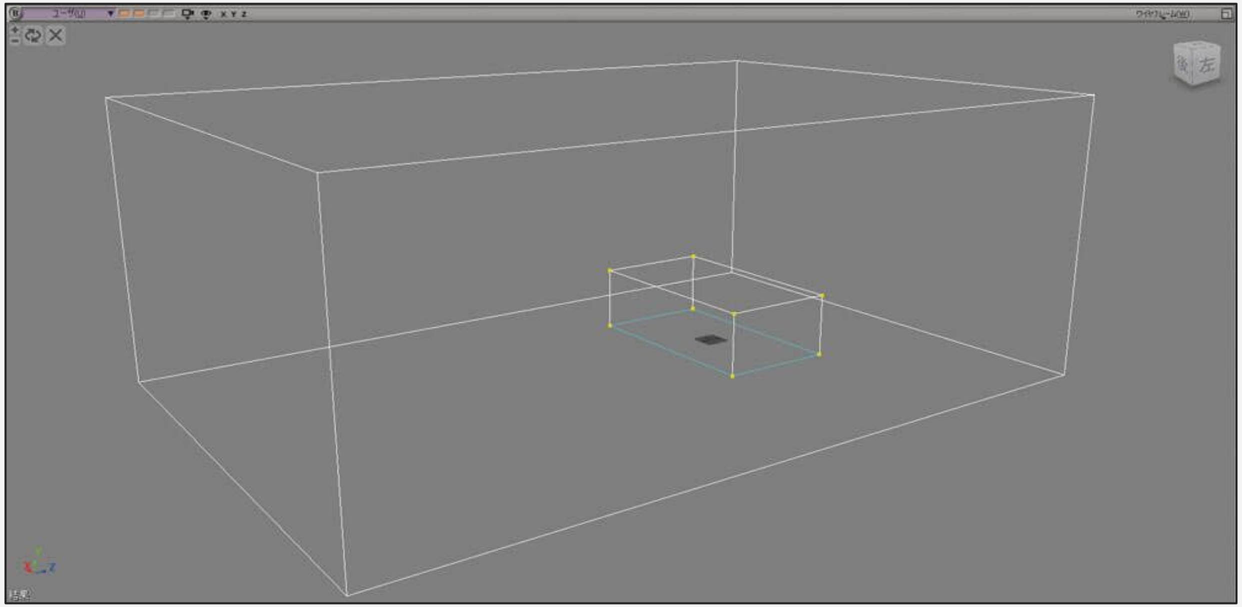
ヘヒョン

やはり既に抑制治療が為されておったウヌとコロニーの住人には調律の異なる部分があるな。

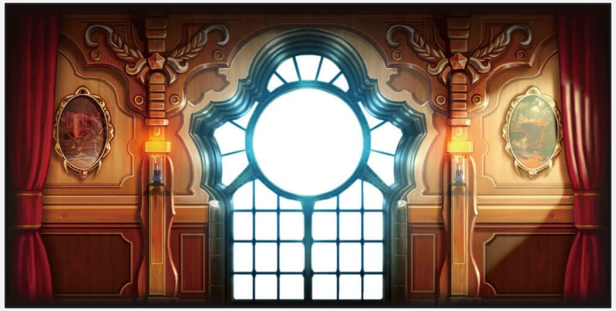
Efficient room creation

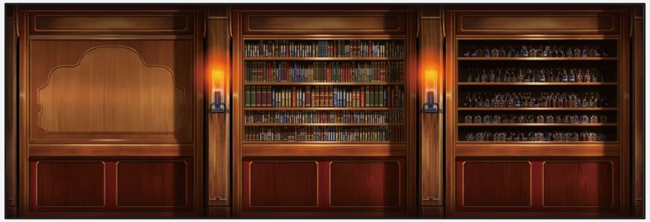
Creating all of the locations in the story mode would have taken too many man-hours, so we established a simple room-type location creation flow for this game. For a basic rectangular room, the model is UV mapped into six pieces, designed, and a background draft is created.

The reference was then given to an outside animation background art studio, who were in charge of painting the walls, floor, and ceiling as seen from directly in front. The texture is then pasted onto the model, adding a sense of atmosphere. If there are any unnatural parts, we correct them and complete the process.

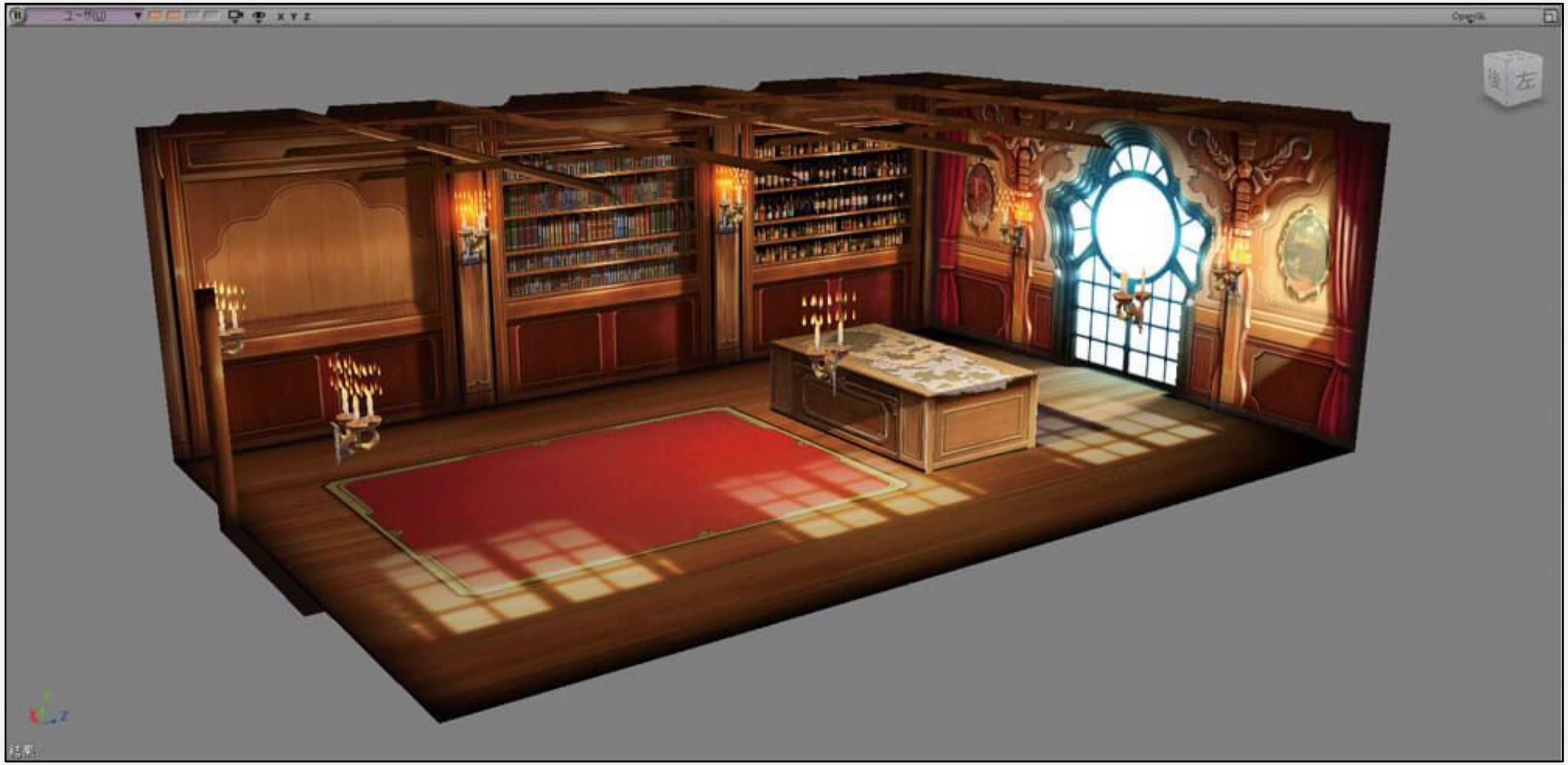


↑ Rectangular model





← Texture Examples ↑

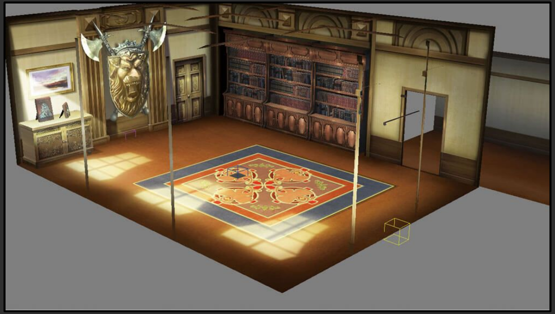


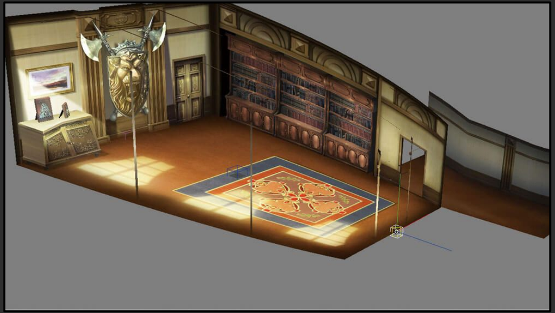
↑ Finished model.

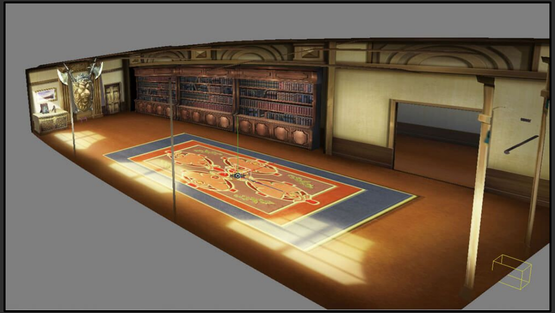
Pseudo angle of view scaling

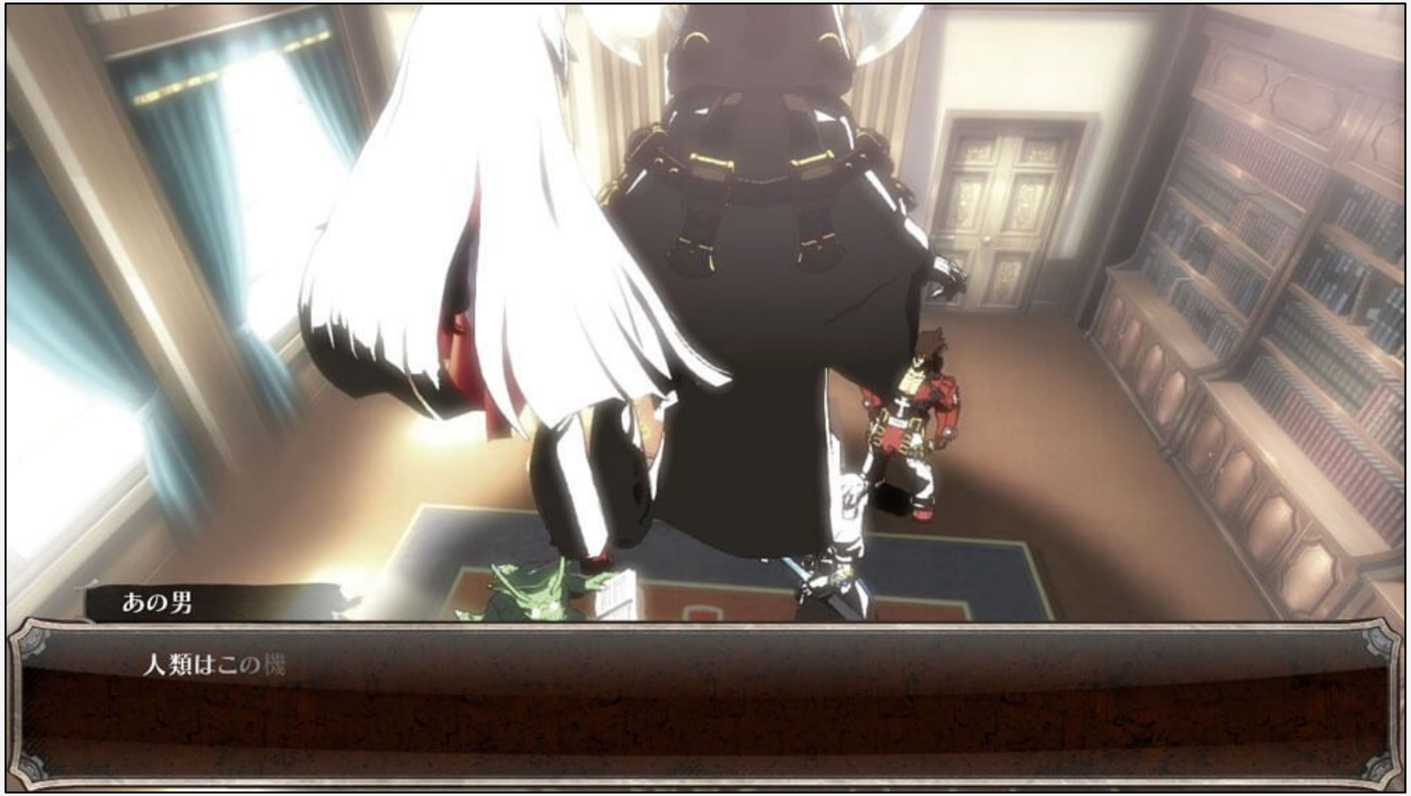
In some cases, the fov value of the camera doesn’t provide enough information about the background or doesn’t produce the intended image.

In order to create the desired compositions, bones are added to the the rectangular model, and the “angle of view” is adjusted by shrinking or stretching the background for each cut. With this, we could create a frame with different angles of view for the characters and the background.


← Standard background.


Example of scaling the foreground. →


← Example of scaling the background.

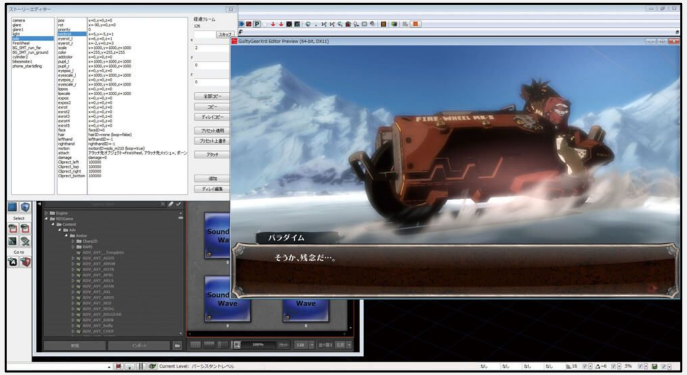


↑ Screenshot. The camera’s angle of view is narrowed to match the characters, while the background is distorted to look more wide-angle.

Story editor

The previous version of the game used a script-based editing system to create story mode cuts, which took a lot of time and effort.

In this title, a dedicated editor has been developed, leading to a significant reduction in workload.


パラダイム

そうか、残念だ…。

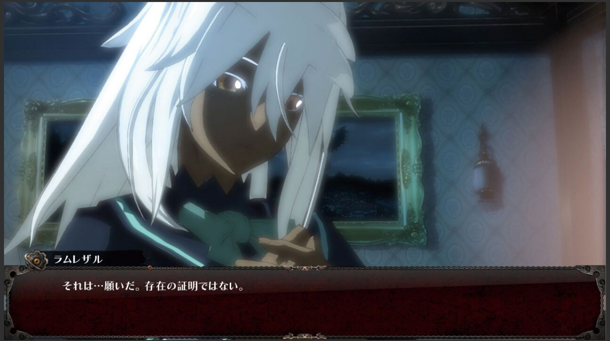
↑ The UI Story Editor has several functions and can be used for almost anything. Can also be controlled with a game controller.

← Bones adjustment function.

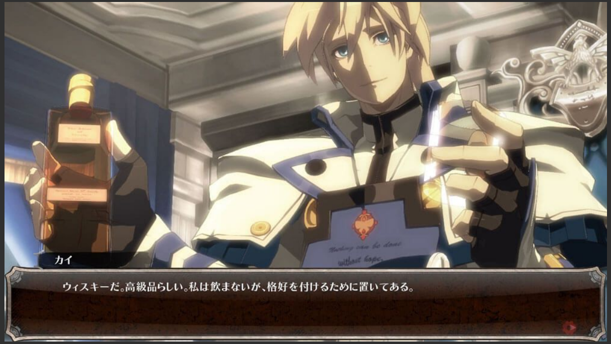
By importing a close-enough pose from the pose library, the bones can be accessed and processed from within the editor.

Look-correction function.

Separate color information can be provided for shaded and lit areas, allowing you to choose colors that match the scene. →


ラムレザル

それは…願いだ。存在の証明ではない。


カイ

ウィスキーだ。高級品らしい。私は飲まないが、格好を付けるために置いてある。

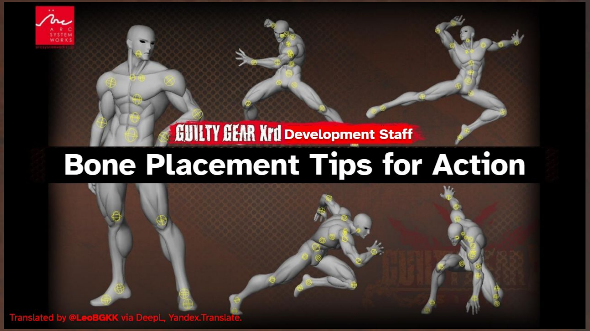
←Attachment function.

Objects can be placed on the character as if they were cel drawings, ignoring 3D space and allowing a more 2D-like effect.

Other GGXrd 3D Materials

GUILTY GEAR Xrd Development Staff
Anime Style Character Modeling TIPS

Translated by @LeoBGKK via DeepL, Yandex.Translate.


GUILTY GEAR Xrd Development Staff
Bone Placement Tips for Action

Translated by @LeoBGKK via DeepL, Yandex.Translate.

GUILTY GEAR Xrd Development Staff
Modeling for Skinning

Translated by @LeoBGKK via DeepL, Yandex.Translate.

The secrets behind
Guilty Gear Xrd -SIGN-'s real-time 3D Anime graphics

This new series, ***"Examining Game Graphics"*** focuses on the graphics of specific titles, and explains the systems and technology behind them. I have been developing ***"Nishikawa Zenji's 3D Game Ecstasy"*** as a series, but since the scope of coverage has become too broad, I would like to deal with technical explanations of specific titles in this new series.
This 1st commemorative issue will feature **GUILTY GEAR Xrd -SIGN-**, a fighting game developed by Arc System Works and rocking arcades since 2014/2.

Original Article .: Writer_西川善司/Zenji Nishikawa ; Photographer_佐々木秀二/Shuji Sasaki , 2014

Translation and slides by .: @LeoBGKK on Twitter
Additional translation by Chev on Polycount Boards

CGWORLD

Non-photorealistic GUILTY GEAR Xrd -REVELATOR- has evolved even further by adding features unique to 3D.

This series has been fascinating players with its 2D anime-like visuals and it's even more impressive with its latest instalment, GUILTY GEAR Xrd -REVELATOR-. This time, let's focus on elements other than the ones we've previously covered (CGWORLD vol.214)

ORIGINAL ARTICLE: TEXT_武田かおり/ KAORI TAKEDA, CGWORLD Vol.215
EDIT__藤井紀明 / Noriaki Fujii（CGWORLD）、山田桃子 / Momoko Yamada

Translation and slides by @LeoBGKK on Twitter

Follow me on twitter.

Leo Bruno "GKK"
@LeoBgkk

🇵🇹 Leonardo Bruno. Garuda. Kanta. DM for COMISSION Info.
✏️Illustration // 📼Video // 🕹️ ButtonMashing.

If you appreciate this translation and wish to donate or something, here's my paypal.
Other presentations will be linked here, as I finish translating them.